

Grado en Ingeniería Telemática
2017-2018

Trabajo Fin de Grado

“Desarrollo de un Recomendador de Videojuegos”

Alejandro Bermejo Vialás

Tutor
Florina Almenares Mendoza

Septiembre 2018



[Incluir en el caso del interés de su publicación en el archivo abierto]
Esta obra se encuentra sujeta a la licencia Creative Commons
Reconocimiento – No Comercial – Sin Obra Derivada

RESUMEN

En la actualidad, las redes sociales forman parte del día a día de la gran mayoría de la población. Las personas han pasado de comunicarse de manera física a una virtual, en la que aparte de interactuar con los demás usuarios, vierten sus opiniones, pensamientos e inquietudes... acerca de cualquier tipo de tema.

Esto hace que se genere un volumen masivo de datos. Datos que pueden resultar de gran utilidad a las empresas para conocer cuál es el estado del mercado en tiempo real.

De ahí la importancia de los sistemas de recomendación, ya que estas herramientas, permiten filtrar y generar resultados, que sirvan a las empresas para generar mayores beneficios.

En este TFG se ha desarrollado un sistema de recomendación, denominado recomendador, de videojuegos basado en filtrado colaborativo, a partir de valoraciones sobre videojuegos extraídas de Twitter. Este recomendador se ha elaborado con el objetivo de ayudar a las empresas a aumentar la venta de videojuegos a través del uso de las recomendaciones que devuelve la herramienta desarrollada.

Palabras clave: Sistema de recomendación, redes sociales, Twitter, videojuegos.

ABSTRACT

Nowadays, online social networks play a great part in the daily lives of a great majority of people. People have moved from physical communication to virtual- in which, apart from interacting with other users, they share their opinions, thoughts and concerns about almost everything, to the masses.

This causes the generation of a massive volume of data, that can be of great use to companies so as to assess the state of the market in real time. This is where recommendation systems come in, since these tools allow filtering and generating results that serve companies to reap far greater benefits.

In this TFG, a videogame recommender based on collaborative filtering has been developed, based on evaluations of video games taken from Twitter. The recommender was developed with the objective of helping companies increase their video game sales through the use of recommendations returned by the tool.

Key words: Recommender system, social networks, Twitter, videogames.

AGRADECIMIENTOS

Con estas palabras, lo que pretendo es expresar mi gratitud a las personas que siempre han estado ahí. Personas que me acompañan día a día y personas que ya no están, pero que siempre tendré presentes en mi camino, un camino que no ha hecho nada más que empezar...

A mi tutora Florina, gracias por entenderme y por toda la comprensión que has tenido conmigo.

A mis padres, Conchi y Félix, gracias por apoyar siempre a vuestra pequeña inversión. Todo es más sencillo cuando se tiene dos puntos de referencia como vosotros.

A mi hermano Alberto, gracias por ser un ejemplo de superación, con ello aprendí que con trabajo y esfuerzo se consigue todo.

Y, por último, lo bueno se hace esperar. A Cynthia, mi compañera en este camino llamado vida, gracias por ser ese gran pilar que me sostiene y por luchar día a día a mi lado.

ÍNDICE

1	INTRODUCCIÓN	6
1.1	Motivación	6
1.2	Medios empleados	6
1.3	Entorno socioeconómico	7
1.4	Marco regulador	8
1.5	Estructura de la memoria	8
2	ESTADO DEL ARTE	9
2.1	Servicios Web	9
2.2	Redes sociales	10
2.2.1	Twitter	11
2.3	Análisis de sentimiento	12
2.4	Sistemas de recomendación	14
2.5	Videojuegos	15
2.6	Tecnologías usadas	16
2.6.1	Java y uso de dependencias Maven	16
2.6.2	Apache Mahout	17
2.6.3	MySQL	17
3	DESCRIPCIÓN DEL SISTEMA	18
3.1	Módulos	19
3.2	Usuarios del sistema	20
3.3	Requisitos específicos	21
3.3.1	Requisitos funcionales	21
3.3.2	Requisitos no funcionales	22
3.4	Casos de uso	23
4	IMPLEMENTACIÓN DEL SISTEMA	26
4.1	Arquitectura software	26
4.2	Clases y Métodos implementados	27
4.3	Funcionalidad del sistema	32
4.4	Interfaz del usuario	37
5	PRUEBAS	40
5.1	Pruebas de interfaz	40
5.2	Pruebas funcionales	45
6	PLANIFICACIÓN Y PRESUPUESTO	49
6.1	Planificación	49

6.2	Presupuesto.....	52
7	CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS.....	54
7.1	Conclusiones	54
7.2	Líneas de trabajo futuras	54

ÍNDICE DE FIGURAS

Figura 2.1 Diagrama ER de la base de datos	18
Figura 3.1 Esquema del sistema completo	19
Figura 3.2 Diagrama de casos de uso del usuario de tipo empresarial.....	24
Figura 4.1 Modelo MVC de la herramienta	27
Figura 4.2 Clases agrupadas en los diferentes componentes de MVC.....	27
Figura 4.3 Diagrama de flujo del Menú principal	33
Figura 4.4 Diagrama de flujo de la extracción de tweets	34
Figura 4.5 Diagrama de flujo de análisis de sentimiento	35
Figura 4.6 Diagrama de flujo de recomendación de videojuegos	36
Figura 4.7 Diagrama de flujo de recomendación orientada a empresa	37
Figura 4.8 Interfaz que representa la pantalla del menú principal del recomendador....	37
Figura 4.9 Interfaz que representa la pantalla para extraer tweets de videojuegos	38
Figura 4.10 Interfaz que representa la pantalla del menú para análisis de sentimiento de los datos.....	39
Figura 4.11 Interfaz que representa la pantalla de recomendación de videojuegos	39
Figura 4.12 Interfaz que representa la pantalla de recomendación orientada a empresa.	40
Figura 5.1 Gráfica de evaluación de recomendaciones en base a threshold vs cálculo de similitud.....	48
Figura 6.1 Diagrama de Gantt del proyecto	51

ÍNDICE DE TABLAS

Tabla 3-1 CU1: Mostrar menú principal	24
Tabla 3-2 CU2: Salir	25
Tabla 3-3 CU3: Extraer tweets	25
Tabla 3-4 CU4: Analizar datos	25
Tabla 3-5 CU5: Recomendar videojuegos.....	25
Tabla 3-6 CU5: Recomendar videojuegos.....	26
Tabla 4-1 Clases del paquete Controller	28
Tabla 4-2 Clases del paquete View - Notifications	28
Tabla 4-3 Clases del paquete View - Views.....	29
Tabla 4-4 Clases del paquete Model - Category & DataAnalysis.....	29
Tabla 4-5 Clases del paquete Model - DBManager	30
Tabla 4-6 Clases del paquete Model - OwnUtils.....	30
Tabla 4-7 Clases del paquete Model - Recommendation	30
Tabla 4-8 Clases del paquete Model - Sentiment	31
Tabla 4-9 Clases del paquete Model - SentimentConfigdata	31
Tabla 4-10 Clases del paquete Model - Tweet	31
Tabla 4-11 Clases del paquete Model - TwitterExtractor.....	31
Tabla 4-12 Clases del paquete Model - User.....	31
Tabla 4-13 Clases del paquete Model - Videogame.....	32
Tabla 4-14 Clases del paquete Model - VideogameTweetStatus	32
Tabla 5-1 T-Uin: Título de la prueba	41
Tabla 5-2 T-UI1:Mostrar menú principal.....	41
Tabla 5-3 T-UI2:Navegar por el menú principal.....	41
Tabla 5-4 T-UI3:Mostrar categorías.....	41
Tabla 5-5 T-UI4:Mostrar videojuegos por categoría.....	42
Tabla 5-6 T-UI5:Mostrar mensaje aviso si no hay videojuegos.....	42
Tabla 5-7 T-UI6:Porcentaje de búsqueda de datos.....	42
Tabla 5-8 T-UI7:Navegación al acabar una extracción de datos.....	43
Tabla 5-9 T-UI8:Mensaje para analizar datos	43
Tabla 5-10 T-UI9:Decisión negativa al analizar datos	43
Tabla 5-11 T-UI10:Mensaje de ausencia de datos para analizar.....	43
Tabla 5-12 T-UI11:Mensaje aviso por agotar recursos	44
Tabla 5-13 T-UI12:Mensaje de ausencia de recomendaciones	44
Tabla 5-14 T-UI13:Listado de recomendaciones de videojuegos	44
Tabla 5-15 T-UI14:Mensaje de aviso ausencia de promotores/detractores	44
Tabla 5-16 T-UI15:Listado de promotores/detractores	45
Tabla 5-17 T-F1:Extracción de tweets	45
Tabla 5-18 T-F2:Matching de usuarios en otros videojuegos	45
Tabla 5-19 T-F3:Nuevos videojuegos (retroalimentación automática).....	46
Tabla 5-20 T-F4:Descarte de tweets duplicados	46
Tabla 5-21 T-F5:Descarte de hashtag duplicados	46
Tabla 5-22 T-F6:Restricción análisis de datos	47
Tabla 5-23 T-F7:Análisis de datos	47
Tabla 5-24 T-F8:Guardar análisis de sentimiento	47
Tabla 5-25 T-F9:Recomendar videojuegos	47
Tabla 5-26 T-F10: Recomendaciones orientadas a empresa	48
Tabla 6-1 Cálculos de amortización	52
Tabla 6-2 Costes de personal.....	53

Tabla 6-3 Presupuesto total 53

1 INTRODUCCIÓN

1.1 Motivación

Un sistema de recomendación es una herramienta de software que nos permite filtrar gran cantidad de información, con el objetivo de crear sugerencias entre los distintos usuarios de la red.

Debido a la gran cantidad de información que se genera en Internet, este tipo de sistemas resulta de gran ayuda a la hora de procesar dicha información, ya que nos permite relacionar los datos relevantes sobre las preferencias de los usuarios y ofrecer una experiencia personalizada en base a su centro de interés.

Se decidió llevar a cabo el desarrollo de una herramienta de recomendación orientada a los videojuegos, debido a la influencia que estos generan en la sociedad y el elevado número de usuarios que se podrían analizar.

En cuanto a los recomendadores de videojuegos, los sitios web que existen sobre este tema, se limitan a realizar preguntas cerradas que condicionan la capacidad del usuario para opinar libremente sobre los videojuegos.

El planteamiento que se propone en el TFG es diferente, ya que se busca generar recomendaciones sobre los usuarios, analizando lo que estos valoran de forma libre y espontánea en una red social, como es Twitter.

En cuanto a las razones personales, cabe destacar en primer lugar el hecho de abordar un trabajo en el que intentar aplicar gran parte de lo aprendido durante el Grado de Ingeniería Telemática. Desde la teoría a la práctica. Por eso, la elección del TFG tiene un enfoque de desarrollo, además de un poco de investigación.

Por otro lado, se pretendió afrontar un tema de actualidad como son las redes sociales y el análisis de información, debido al impacto que tienen éstas en el mundo en el que vivimos. Esto implica el hecho de tener que probar nuevas tecnologías, que conllevan nuevos retos tecnológicos, como son los sistemas de recomendación, o recomendadores. El funcionamiento de estos sistemas, aunque están muy presentes en el día a día cuando navegamos por la red, no es bien conocido para la mayoría de usuarios.

Por último, destacar que la elección de que el sistema de recomendación esté enfocado en el mundo de los videojuegos, por un lado, se debe al mercado y cantidad de usuarios que participa en este sector y, por el otro lado, representa una gran motivación personal.

1.2 Medios empleados

El siguiente listado, muestra los medios utilizados para llevar a cabo el proyecto, tanto a nivel de desarrollo como para la documentación del mismo, desglosados en los apartados de hardware y software.

- **Hardware**

- Apple Macbook Pro 13'. Utilizado para el desarrollo de la herramienta de recomendación y su documentación.

- **Software**

- Eclipse Oxygen/ IntelliJ IDEA. Los entornos de desarrollo.
- Apache Mahout. Empleado por su implementación del algoritmo de recomendación.
- Librería twitter4j. Utilizada porque permite manejar la *API Search* de Twitter, para extraer datos.
- Aylien. Uso de su API de Análisis de Sentimiento, para analizar los datos extraídos de Twitter.
- DBeaver/SequelPro. Utilizado para gestionar la base de datos MySQL.
- Bitbucket. Repositorio del código y gestión de versiones.
- Microsoft Office 365. Paquete ofimático empleado para documentar (Word) y para visualizar ficheros de datos (Excel).

1.3 Entorno socioeconómico

El año 2017, ha sido muy favorable para el crecimiento de la economía española. Según el INE, el año pasado España creció un 3,1%, un valor superior al del resto de países de la Unión Europea.

El FMI ha estimado que la previsión de crecimiento durante 2018 será de un 2,8%, un dato bastante positivo para todos los sectores, entre ellos, el sector de los videojuegos, que en el último año experimentó un aumento en la facturación de un 16,9% respecto de 2016.

Esto sitúa a la industria de los videojuegos como líder en el sector de ocio audiovisual del país. Actualmente, genera más de 8.000 empleos directos en España y cada uno de ellos, a su vez, 2'6 empleos indirectos [1].

Todos estos datos indican, que la industria del videojuego, es un negocio al alza en el que aparte de incrementarse los beneficios, se generan puestos de trabajo que ayudan a nuestra sociedad a crecer.

A todo este éxito, se le suma la repercusión social que los videojuegos generan en redes sociales, plataformas de *streaming* de video, etc. En la actualidad, no es de extrañar ver a todo tipo de personas en redes de contenidos, como puede ser *Youtube*, compartiendo cómo juegan, trucos que faciliten ganar logros o comentando retransmisiones... en busca del reconocimiento del resto de la comunidad de jugadores, sin importar tu edad, sexo o raza, ya que lo único que importa, es la habilidad que se tiene en un determinado videojuego.

1.4 Marco regulador

En esta sección, se describe la normativa a la que está sujeta la herramienta desarrollada en el TFG.

Por un lado, al manejar datos de usuarios extraídos de la red social Twitter, y en general, cuando se manejan datos de tipo personal, estamos sujetos a la LOPD, Ley Orgánica de Protección de datos [22], BOE núm.98, de 14 de diciembre de 1999. Esta ley, tiene como objeto garantizar y proteger, en lo que concierne al tratamiento de datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y en especial, su honor e intimidad.

Aunque cabe destacar, que en la herramienta nunca se usan datos de carácter privado, como podrían ser teléfonos, datos bancarios, etc.

Recientemente, desde el 25 de mayo de 2018, se debe aplicar la directiva GDPR, Reglamento General de Protección de Datos [23], que es de debido cumplimiento a nivel europeo. Este reglamento afecta a cualquier empresa que maneje información personal de cualquier tipo y que realice negocios en la Unión Europea.

Con vistas a su distribución al público en un futuro, la aplicación está sujeta a la LSSI, Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, BOE núm. 166, de 12/07/2002 [25].

Esta ley establece las reglas necesarias, a proveedores de servicios, empresas que ofrecen productos o ciudadanos que posean páginas web, para que el uso y disfrute de esta red sea segura, confiable y ofrezca una experiencia positiva [26].

Además, la herramienta se encuentra bajo la Ley de Propiedad Intelectual, ya que dicta que la propiedad intelectual de una obra literaria, artística o científica corresponde al autor por el mero hecho de su creación. Esta ley se encuentra definida en el BOE núm. 97, de 22/04/1996 [27].

En cuanto a las herramientas que se han utilizado para llevar a cabo el proyecto, la mayoría son “open source” o se ha usado la capa gratuita que ofrecían. En el caso del entorno de desarrollo, IntelliJ, y de Microsoft Office, se posee licencia de estudiante, por lo que no se infringe ninguna normativa de uso.

1.5 Estructura de la memoria

En esta sección, se incluye un breve resumen de cada capítulo, para que sea más fácil seguir

- Capítulo 1, "Introducción": Se realiza la descripción de los objetivos del proyecto, motivación, los medios empleados para su realización, así como el marco regulador y el entorno socioeconómico.

- Capítulo 2, "Estado del arte": Se describen las tecnologías utilizadas en el proyecto, así como la definición sobre sistemas de recomendación, redes sociales y videojuegos.
- Capítulo 3, "Descripción del sistema": Engloba todo lo relacionado con la definición del sistema, es decir, los módulos que lo componen, requisitos funcionales y casos de uso.
- Capítulo 4, "Implementación del sistema": Se describe la arquitectura software de la herramienta, diagramas de flujo para entender la lógica del recomendador. También se incluye la representación de las diferentes pantallas de la aplicación.
- Capítulo 5, "Pruebas": Se describen las pruebas realizadas al sistema, así como el resultado obtenido en ellas.
- Capítulo 6, "Planificación y presupuesto": Muestra la planificación del proyecto y el cálculo del coste que supondría.
- Capítulo 7, "Conclusión": En este capítulo, se exponen las conclusiones obtenidas y las posibles líneas futuras de desarrollo del proyecto.
- Anexo, "Glosario": Glosario de los términos.

2 ESTADO DEL ARTE

En este capítulo se describen las tecnologías utilizadas para llevar a cabo este Trabajo Fin de Grado.

2.1 Servicios Web

La palabra REST (*Representational State Transfer*), Transferencia de Estado Representacional, es un término que fue acuñado por Roy Fielding, en una tesis doctoral sobre la web en el año 2000 [10].

Este término se utiliza para describir las interfaces entre sistemas, que usan HTTP para obtener datos o ejecutar operaciones sobre ellos.

En REST, la implementación del cliente y el servidor, se pueden realizar de forma independiente. Si cada uno de ellos conoce el formato de los mensajes que debe enviar al otro, podemos mantener una estructura separada por módulos. Esto nos proporciona una gran ventaja, ya que, si realizamos modificaciones en el cliente, no afectan al lado servidor.

Otra de las ventajas, es el hecho de separar la interfaz de usuario respecto al almacenamiento de datos. Nos otorga mayor flexibilidad y mejora la escalabilidad. Al utilizar una interfaz REST, todos los clientes pueden preguntar al mismo servicio, realizar las mismas acciones y recibir las mismas respuestas.

REST sigue un paradigma *stateless*, sin estado. ¿Qué quiere decir esto? REST permite que un servidor no necesite saber el estado de los clientes, ni que los clientes sepan el estado del servidor. Este modelo, permite al servidor y al cliente entender cualquier mensaje que reciban, sin necesidad de conocer los mensajes que se intercambiaron en el pasado.

Cuando se emplea REST, las peticiones contienen una URI, que es una ruta única hacia un recurso. Estas rutas, deben definirse de tal manera que el cliente sea capaz de reconocer fácilmente su cometido.

El uso de estos recursos, en vez de utilizar comandos, proporciona a las aplicaciones REST confiabilidad, buen rendimiento y un grado alto escalabilidad [11].

Los datos que se generan en la respuesta de un servicio REST, suelen ser en formato XML o JSON. Esto permite que los servicios REST sean independientes del tipo de plataforma o lenguaje que se esté usando para desarrollar.

Para este TFG, se han utilizado dos API REST. Una de ellas, para buscar/extraer tweets (API Search) de Twitter, la otra, para realizar análisis de sentimiento (Aylien) de cada uno de los tweets encontrados, tal como se describe a continuación.

JSON

JSON(JavaScriptObjectNotation) es un formato ligero de intercambio de datos, que resulta muy fácil leer y escribir a los humanos. Para las maquinas, es sencillo pasarlo y generarlo.

JSON está basado en un subconjunto del lenguaje de programación Javascript. Su formato es texto plano y resulta familiar a los programadores, debido a su parecido con otros lenguajes como C, C++, Javascript, Java o Python. Por eso, lo convierte en un lenguaje ideal para el intercambio de datos.

Con este tipo de formato, se pueden construir colecciones de par nombre/valor, lo que comúnmente en otros lenguajes representaría objetos y listas ordenadas de valores.

2.2 Redes sociales

Las redes sociales son sitios web constituidos por conjuntos de personas u organizaciones de diversas características o intereses, que, a partir de estas, intercambian información [30].

Desde que el ser humano existe ha establecido relaciones grupales (familia, amigos, trabajo, etc.), por lo que el hecho de que los individuos se agrupen, en este caso en la red, es natural para nosotros.

Para la gran mayoría, las redes sociales son consideradas un servicio novedoso en internet [32], debido a la época de máximo esplendor en los últimos años, pero lo cierto es que el origen de las redes sociales se sitúa alrededor del año 1995, cuando Randy Conrads creó classmates.com. Esta red, permitía recuperar o mantener el contacto con los compañeros de la universidad. Años más tarde, en 2002, empiezan a crearse sitios web donde poder

establecer un círculo de amigos en línea, como MySpace, y como no, Facebook, fundado en 2004.

Se pueden encontrar redes sociales de todo tipo, desde redes sociales genéricas, donde las personas conectan con sus familiares y amigos; redes profesionales como LinkedIn, que permiten compartir información sobre diferentes sectores o aumentar las posibilidades laborales, y redes sociales temáticas, creadas para intercambiar información sobre un contenido en concreto, por ejemplo, cine, música o videojuegos [31].

En la actualidad, el impacto que suponen las redes sociales en la sociedad es tan grande, que el análisis de estas se ha convertido en parte de estudio de la sociología o la antropología, y las grandes empresas analizan todo lo que en ellas se comenta para conocer el estado del mercado y predecir posibles cambios en él.

Para este proyecto, se contempló la extracción de datos de redes sociales de temática *gamer*, como Twitch, Raptr y WeGame entre otras, para posteriormente analizar lo que los usuarios valoraban sobre videojuegos e intentar generar recomendaciones en base a esas valoraciones, pero dichas redes son muy cerradas, no disponen de APIs con las que interactuar y eso entorpece el trabajo que se quiere llevar a cabo.

Por esta razón, se tuvo que poner la vista en redes sociales abiertas, que disponen de servicios que permiten consultar la información que se comenta dentro de ellas.

2.2.1 Twitter

La red social que se ha elegido es Twitter. Esta red social, fue fundada en 2006 por Jack Dorsey [33].

El funcionamiento de Twitter es muy sencillo, te permite escribir mensajes de un máximo de 280 caracteres (antes 140), denominados “*tweets*”, de forma pública.

Si a una persona le gusta lo que publica otra persona, puede suscribirse a sus tweets, lo que comúnmente en Twitter se llama “seguir” y a las personas se les denomina “seguidores” o “*followers*”.

En ella 320 millones de usuarios mensuales [34], comentan sobre todo tipo de temas. Esto, junto con su carácter público, convierte a Twitter en una fuente de información muy valiosa para las empresas, ya que les permite conocer qué opina la gente sobre su marca en tiempo real.

Viendo ese gran potencial, se ha elegido esta red social para extraer datos sobre videojuegos, que podamos analizar. Además, Twitter pone a disposición de los desarrolladores APIs que permiten consultar toda la información, por lo que facilita bastante el trabajo. En concreto, en este TFG, se usa la API (*Application Program Interface*) Search de Twitter [12], como se explica a continuación.

API Search de Twitter

La API Search de Twitter, permite buscar y extraer colecciones de tweets que coinciden con una consulta realizada.

Este servicio responde con una estructura JSON que representa objetos *tweet* y nos permite leer la información de todos aquellos que devuelva el servicio. En el proyecto, se lee el nombre público del usuario en Twitter, es decir, su *Nick*; el texto del tweet, los *hashtag* que contiene el tweet y el identificador único del tweet.

El recurso que pone Twitter a disposición de los desarrolladores para este cometido es <https://api.twitter.com/1.1/search/tweets.json>.

Esta API tiene una serie de restricciones de uso definidas por Twitter.

- Es obligatorio el uso de autenticación en las peticiones, en el caso de este TFG se trata de autenticación por aplicación, y dependiendo del tipo de autenticación, se permite un número mayor o menor de peticiones.
- Si se realiza por aplicación, son 450 peticiones por ventana temporal de 15min, mientras que, si la autenticación es por usuario, Twitter sólo permite 180 peticiones cada 15 min.

Para realizar las consultas al servicio REST, se ha utilizado una librería externa, llamada *twitter4j*.

Twitter4j es una librería de código abierto Java que implementa la API de Twitter [13][14]. Se puede utilizar cuando se desarrolla en lenguaje Java gracias a su dependencia Maven, permitiendo importar todo su código al proyecto. Aunque no es una librería oficial de Twitter, gracias a ella, se integran de forma rápida y sencilla los métodos para comunicarse con la API Search de Twitter y llevar a cabo las peticiones que se necesitan en el TFG.

```
<dependency>
  <groupId>org.twitter4j</groupId>
  <artifactId>twitter4j-core</artifactId>
  <version>4.0.6</version>
</dependency>
```

2.3 Análisis de sentimiento

Una vez que los datos se extraen de alguna fuente de información, por ejemplo, redes sociales, requiere su análisis para la obtención de resultados. En este caso, se hace un análisis de sentimiento de los “tweets” extraídos que permitirá valorar un comentario.

El análisis de sentimiento es un proceso dentro del *data mining* (minería de datos), que se utiliza para intentar determinar el tono emocional que tiene una frase [7], documento o en el caso de este proyecto, un tweet. Lo que se intenta, es saber si la persona que escribió el texto, refleja una actitud positiva o negativa ante un tema concreto.

El uso de este tipo de análisis de datos, se ha vuelto muy útil para las empresas [8], ya que resulta muy sencillo, conocer la opinión que tienen las personas sobre los productos o servicios que ofrecen, de esta forma, son capaces de reaccionar ante las necesidades del mercado con mayor rapidez.

En este TFG, el análisis de sentimiento, se emplea para concretar si los usuarios de Twitter hablan bien o mal de los videojuegos sobre los que comentan en esta red social. Esto nos ayuda a generar las tablas de entrada que el software de recomendación es capaz de procesar. Tablas en las que relacionamos a un usuario con el videojuego sobre el que ha opinado y su grado de satisfacción de este mismo.

Se pueden encontrar diversas soluciones, que proveen la posibilidad de realizar análisis de sentimientos mediante APIs, entre ellas se encuentran Azure o MeaningCloud, pero en el proyecto utilizaremos la solución proporcionada por Aylien.

API Text Analysis de Aylien

La API Text Analysis de Aylien, permite entre otras cosas, realizar análisis de sentimiento.

Aylien, es una empresa que da, a otras empresas y a desarrolladores, la capacidad de recopilar, analizar y comprender contenido generado por las personas.

A diferencia de otros actores del sector, Aylien, posee un “modo tweet” específico dentro de su API de análisis de sentimiento.

Aylien, provee a los desarrolladores una dependencia Maven [16], para que la integración con su servicio sea sencilla, debajo se encuentra la utilizada en el TFG.

```
<dependency>
  <groupId>com.aylien.textapi</groupId>
  <artifactId>client</artifactId>
  <version>0.6.0</version>
</dependency>
```

La API devuelve como respuesta una estructura JSON como la siguiente:

```
{
  "polarity": "positive",
  "subjectivity": "subjective",
  "text": "John is a very good football player",
  "polarity_confidence": 0.9999936601153382,
  "subjectivity_confidence": 0.9963778207617525
}
```

Para el proyecto, sólo se utilizan los campos: *polarity* y *polarity_confidence*, de la respuesta [15], ya que son los que nos proporcionan la información relevante para poder crear la entrada al sistema de recomendación.

2.4 Sistemas de recomendación

Un recomendador, es un sistema que se encarga de, como su nombre indica, recomendar o sugerir a los usuarios ítems o productos, en base a sus preferencias.

Los sistemas de recomendación, son de dos tipos:

- **Basados en contenido**

En los recomendadores basados en contenido [36], el usuario recibe información similar a la que haya mostrado interés. El producto es la base de la predicción. ¿Cómo funciona? Lo que se utiliza para establecer las relaciones son las características del ítem en cuestión, por ejemplo, categoría, precio, tamaño...)

- **Filtrado colaborativo**

Este tipo de recomendadores, basa su lógica en las características de cada usuario [37]. El sistema analiza sus preferencias, es decir, cómo ha valorado un determinado producto y los compara con otros usuarios que valoraron productos parecidos.

Dentro de este tipo de sistemas de recomendación, se pueden definir dos vertientes, la primera, *user-based-recommender*, que se fija en los usuarios y la similitud entre ellos, la segunda, *item-based-recommender*, igual que la anterior, pero se centra en el ítem en sí y los ítems parecidos que hay en el sistema.

En este TFG, usaremos filtrado colaborativo y específicamente, *user-based-recommender*, de tal manera que extraeremos las opiniones de los usuarios de Twitter sobre determinados videojuegos, para tratar de generar recomendaciones.

En el filtrado colaborativo, un factor muy importante es la similitud. Ya que determinará el grado en el que un usuario se parece a otro y por tanto, si se le debe recomendar un ítem que otro usuario valoró o no.

Hay diferentes modos de calcular la similitud entre usuarios. Dentro de Mahout (apartado 2.5.2.) se encuentran implementados los métodos más importantes, como son:

- **Coeficiente de Correlación de Pearson**

El coeficiente de correlación de Pearson mide el grado de relación de dos variables, en este caso, mide la tendencia de los valores de los usuarios a moverse juntos. La correlación toma valores entre -1 y 1 [40].

El cálculo de la correlación de Pearson se realiza de la siguiente manera (Ecuación 2.1):

$$r_{xy} = \frac{\sum z_x z_y}{N} \quad (2.1) [39]$$

El problema de este cálculo, aplicado a nuestro caso práctico, es cuando no existen videojuegos en común, ya que no se podrá calcular la similitud y por tanto, no existirán recomendaciones.

- **Distancia Euclídeana**

Este método, se basa en el cálculo de la distancia entre usuarios (Ecuación 2.2). Se representa a los usuarios como coordenadas [29] de preferencias sobre los diferentes videojuegos.

$$d(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2} \quad (2.2)$$

- **Coseno**

El cálculo de la similitud mediante el coseno (Ecuación 2.3), representa los usuarios y sus preferencias como vectores. De esta manera, el ángulo que forman entre ellos, no dará la idea de lo parecidos que son o no. Cuanto menor sea el ángulo, más se aproxima el valor a uno, y más afines serán los usuarios, si los vectores son ortogonales el coseno se anula, y si tienen un sentido contrario, el valor sería -1 [29].

$$\cos(x1, x2) = \frac{x1 \cdot x2}{\|x2\| \Delta \|x1\|} \quad (2.3)$$

- **Correlación de Spearman**

Se trata de una variante de la correlación de Pearson, solo que, en vez de calcular la similitud en base a las preferencias, realiza un ranking relativo de preferencias. Supone mucho coste computacional, por lo que no se suele utilizar [29].

- **Coefficiente de Tanimoto**

El coeficiente de Tanimoto, o coeficiente de Jaccard, ignora el valor de las preferencias, representa el número de ítems para el que dos usuarios han hecho una valoración, independientemente de si es buena o mala [29].

2.5 Videojuegos

Si buscamos la palabra videojuego, encontramos la siguiente definición según la RAE:

“Juego electrónico que se visualiza en una pantalla” [2]

Pero lo cierto es que el mundo del videojuego engloba mucho más a día de hoy.

Ha pasado más de medio siglo, desde que en la década de los 50, aparecieran los primeros videojuegos, poco tiempo después de que llegasen las primeras computadoras. Aunque no fue hasta los años 70, con el descenso de los costes de fabricación, cuando se lanzaron los primeros videojuegos dirigidos al gran público [4].

Al principio, los videojuegos se jugaban en máquinas recreativas, pero ahora hay un gran abanico de posibilidades donde poder disfrutar de ellos; PC, videoconsolas (PlayStation, Wii, etc.), teléfonos móviles, gafas de realidad virtual, tabletas...

Desde entonces, la industria de los videojuegos, se ha convertido en una de las mayores a nivel de facturación. Sólo en España, esta industria facturó 1.359 millones de euros en

el año 2017, un 16,9 % más que en el año 2016, según indica el Anuario 2017 de AEVI [1] [5].

El mundo de los videojuegos se ha convertido en un fenómeno social. Lejos queda ya el tiempo en que los juegos eran para un solo jugador o como mucho dos, todo eso ha cambiado, hoy en día se puede acceder a plataformas donde millones de personas juegan, se relacionan y comparten objetivos para conseguir una meta.

Ahora, es algo normal ver a personas compartiendo en redes sociales cómo jugar a determinados videojuegos o cómo conseguir logros especiales dentro de ellos. Se ha convertido en una industria capaz de mover a las masas, para llenar estadios donde se disputan competiciones de jugadores profesionales, que han convertido el entretenimiento en una forma de ganarse la vida. Existen redes sociales específicas sólo para hablar de videojuegos o plataformas de video en *streaming*, que ofrecen en exclusiva retransmisiones de competiciones mundiales de LoL, Clash Royale...

Por todo esto, es un mercado muy bueno en el que aplicar análisis y procesamiento de los datos, para aportar aún más valor a las empresas que necesitan saber lo que se opina sobre sus videojuegos y que sean capaces de adaptarse a cambios de una manera ágil.

2.6 Tecnologías usadas

En esta sección, se habla de todas las tecnologías usadas para desarrollar el proyecto, lenguaje de programación, servicios REST, MySQL y el software para recomendar videojuegos.

2.6.1 Java y uso de dependencias Maven

Java es un lenguaje de programación orientado a objetos [18], desarrollado por James Gosling en 1982 para Sun Microsystems. Publicado en 1995, Java se ha convertido en un componente fundamental a la hora de desarrollar.

La principal característica de este lenguaje de programación es que es orientado a objetos. ¿Qué quiere decir esto?, permite crear entidades genéricas, llamadas objetos, que se puedan reutilizar entre diferentes proyectos. Es decir, estos objetos se comportan como piezas de un puzzle que se pueden volver a usar y que reducen el tiempo de desarrollo de un proyecto al no tener que empezar siempre desde cero [17].

Otra característica muy importante es la independencia de la plataforma. Los programas Java pueden ejecutarse en cualquier máquina, por eso Sun Microsystems creó el eslogan “*Write once, run anywhere*” [19], “*Escribe una vez, ejecútalo en cualquier parte*”.

Para este TFG, no sólo se ha utilizado Java, también se ha usado una potente herramienta de administración de dependencias llamada Maven [20].

Maven es una herramienta open-source creada con el objetivo de simplificar el proceso de construcción del código. Para llevar a cabo esa construcción, Maven utiliza un POM

(Project Ibet Model), un fichero XML de configuración, que describe el proyecto que se va a construir, las dependencias y los componentes externos [21].

Gracias a la gestión que realiza Maven con el POM, basta con definir las dependencias necesarias en el proyecto, para hacer uso de los objetos y métodos implementados en librerías externas.

En este proyecto, se han usado dependencias Maven para integrar la librería twitter4j (apartado 2.2.1), los servicios de Aylien para análisis de texto (apartado 2.3) y el sistema de recomendación de Apache Mahout (apartado 2.6.2).

2.6.2 Apache Mahout

Apache Mahout comenzó su andadura en el año 2008 como subproyecto de Apache Lucene, el reconocido motor de búsqueda *open source*, que proporciona implementaciones de búsqueda avanzada, *text mining* y técnicas de recuperación de información.

Mahout es una librería *open source* de *machine learning*, que implementa algoritmos de aprendizaje automático. Los algoritmos que contiene son sobre motores de recomendación (filtrado colaborativo, apartado 2.4.), clustering y clasificación. Está implementada en Java y pensada para desarrolladores, ya que carece de interfaz de usuario o de un instalador [28]. En el TFG se ha integrado utilizando la dependencia Maven, que se muestra debajo.

```
<dependency>
  <groupId>org.apache.mahout</groupId>
  <artifactId>mahout-core</artifactId>
  <version>0.9</version>
</dependency>
```

Esta librería, es la pieza más importante del puzzle, en este proyecto, ya que nos permite procesar los datos resultantes, después de realizar el análisis de sentimiento, para generar las posibles recomendaciones de videojuegos a cada usuario.

2.6.3 MySQL

MySQL, es el sistema de gestión de bases de datos relacional considerado como el más popular en todo el mundo. En sus inicios, fue desarrollado por MySQL AB, una de las empresas más grandes de software libre del mundo, pero en 2008 la empresa fue adquirida por Sun Microsystems, y dos años después, esta fue comprada por Oracle, actual propietaria del software [6].

La licencia de MySQL es dual, ya que si se quiere usar de manera particular es gratuita. En cambio, si se quiere utilizar para desarrollar productos empresariales donde se incluye MySQL dentro de la propia solución software, se debe pagar una licencia.

En este proyecto, se ha utilizado MySQL por su simplicidad a la hora de montar la estructura del modelo, por la flexibilidad que te provee el uso de tablas relacionales y porque nos permite realizar transacciones, definir claves foráneas e indexar los datos.

Con MySQL resulta muy sencillo manejar toda la estructura de datos de nuestro modelo, ya que las consultas se asemejan mucho al lenguaje natural de las personas.

Además, es un software multiplataforma que nos permite utilizarlo del mismo modo independientemente del sistema operativo en el que estemos desarrollando, ya sea Linux, Mac Os X o Windows, entre otros muchos soportados por MySQL.

En el desarrollo del recomendador, hemos creado seis tablas para la persistencia de los datos. Cinco representan las entidades *Sentiment*, *Tweet*, *User*, *Videogame* y *Category*, y una almacena los campos de configuración de la API de análisis de sentimiento.

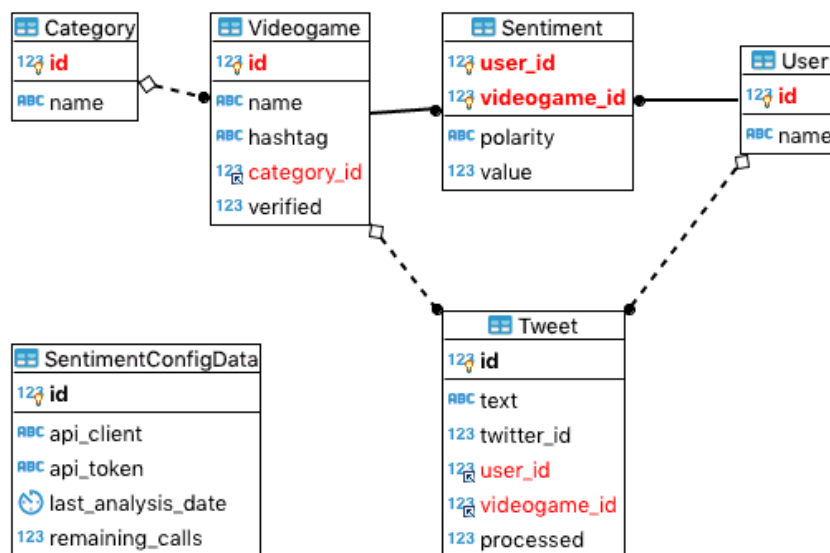


Figura 2.1 Diagrama ER de la base de datos

3 DESCRIPCIÓN DEL SISTEMA

En este capítulo se describe el sistema que se ha desarrollado en el TFG, los módulos que lo componen, los requisitos funcionales que cumple en su implementación y los casos de uso de la herramienta.

En la Figura 3.1. se muestra la arquitectura del sistema, la cual contiene un módulo de extracción de datos, en este caso, extraemos información desde Twitter. Un segundo módulo de análisis, que procesa los datos mediante uso de análisis de sentimiento y por último, en el tercer módulo, donde se producen las recomendaciones tanto de videojuegos, como de usuarios por categoría.

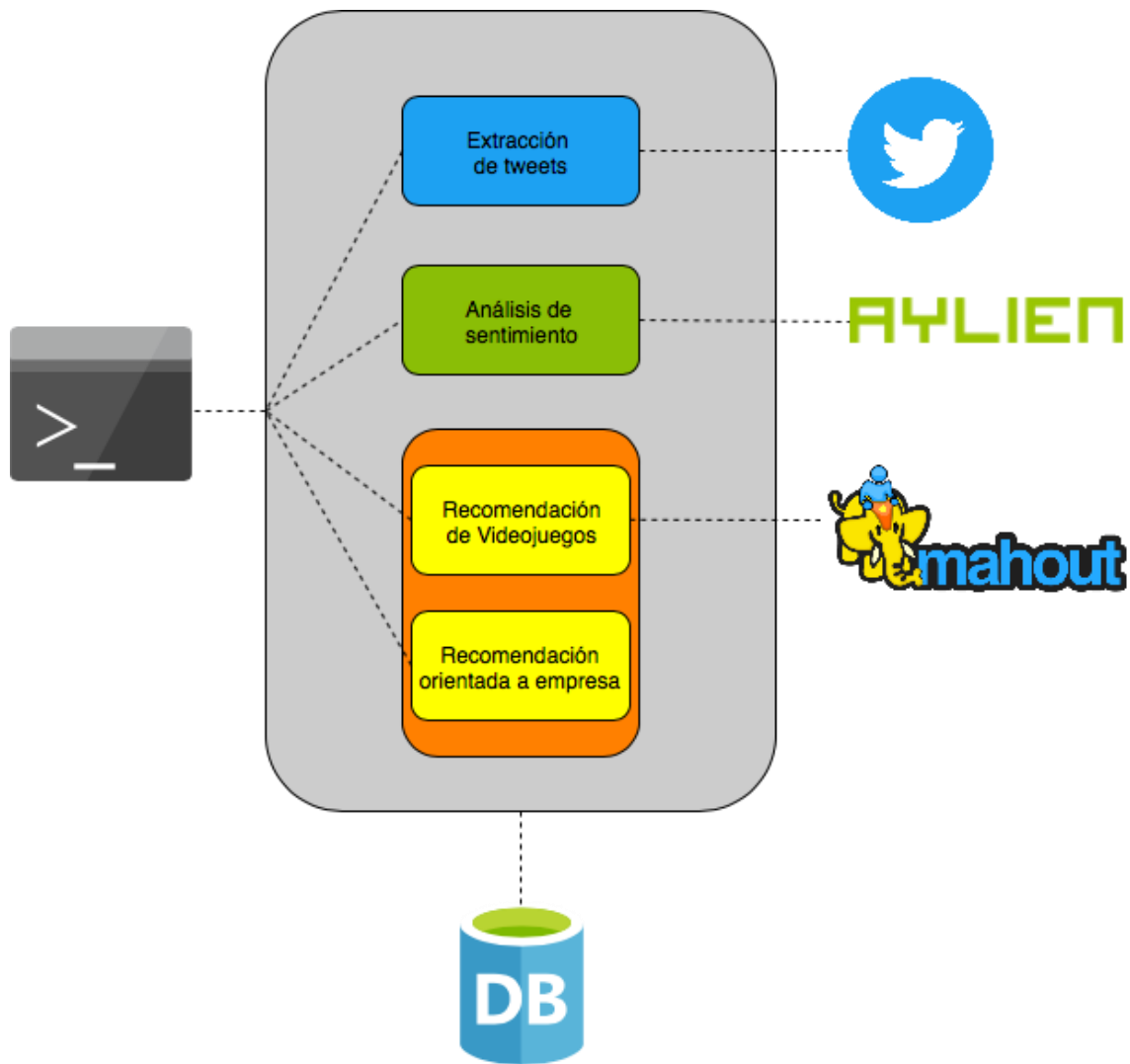


Figura 3.1 Esquema del sistema completo

El sistema desarrollado es capaz de extraer datos sobre videojuegos de Twitter, analizarlos mediante proceso de análisis de sentimiento y generar recomendaciones en base a esos datos.

Ha sido necesario incluir en el sistema un cliente de la API Search de Twitter, para extraer los datos. Un cliente de la API Text Analytics de Aylien, para realizar análisis de sentimiento. Y también, un servidor MySQL en el que se persisten los datos de la herramienta implementada.

En el siguiente apartado se describe más en detalle cada uno de estos módulos.

3.1 Módulos

En esta sección se describen los módulos de la aplicación que forman el sistema completo.

La herramienta desarrollada en este TFG se divide en tres módulos principales, en base al cometido que desempeñan dentro del sistema. Esto nos permite añadir, reemplazar o

modificar funcionalidad dentro de cada uno de ellos sin necesidad de realizar cambios, o efectuando cambios mínimos, en el resto del sistema.

- El primer módulo, es el que se usa para la extracción de datos o cómo fuente de entrada de la información. En este caso, utilizamos cómo entrada de datos la red social Twitter, de la que extraemos tweets, pero podríamos añadir otros formatos de entrada, como por ejemplo ficheros CSV con la información de entrada, etc.

Gracias a este módulo, obtenemos los datos que serán procesados por el módulo de análisis.

- En el segundo, el módulo de análisis, procesamos los datos para generar entradas que el recomendador entienda. Para este proyecto, lo que hemos hecho es implantar un módulo de análisis de sentimiento, que nos permite evaluar la satisfacción de los usuarios con respecto a los videojuegos. Estas valoraciones van de 1 a 5, siendo 1 el grado más negativo y 5 el más positivo.

Como se comentaba anteriormente, a partir de aquí, podemos generar una entrada que el recomendador entiende. Estas entradas siguen la siguiente estructura: “usuario, ítem, valor”.

- Por último, en el tercer módulo es el de recomendación, que incluye dos funcionalidades.
 - La primera, realiza recomendaciones de videojuegos para cada usuario, y la segunda, devuelve recomendaciones de usuarios en base a la categoría y a cómo evaluaron los videojuegos.
 - La segunda, la hemos denominado “recomendación orientada a empresa”. Esta funcionalidad, recomienda usuarios promotores y detractores de cada categoría.
Un usuario promotor, entendiendo promotor como alguien que defiende la marca, en este caso, que promueve una categoría de videojuegos, será aquel que como sentimiento tenga un valor mayor a 3.
Del mismo modo, un usuario promotor será aquel con valoración sobre un videojuego menor a 3.

3.2 Usuarios del sistema

Para este TFG se ha decidido tener dos tipos de usuarios, uno será el administrador/desarrollador, y el otro el usuario de la herramienta.

- Administrador/desarrollador

Este usuario es el encargado de mantener el sistema, así como de generar nuevas versiones del mismo o implementar posibles modificaciones para mejorar su funcionamiento.

- Usuario de tipo empresarial

Es cualquier persona dentro del ámbito empresarial del sector de los videojuegos, que adquiera la aplicación para uso profesional, orientado a la mejora de los beneficios de su empresa y a la retención o logro de nuevos clientes.

3.3 Requisitos específicos

En este capítulo, se describen los requisitos específicos de la herramienta, tanto requisitos funcionales como no funcionales.

3.3.1 Requisitos funcionales

Los requisitos funcionales detallan los servicios que el sistema debería proporcionar[9]. Representan, la manera en la que el sistema debe comportarse dependiendo de las entradas particulares al mismo y cómo actúa en determinadas situaciones. Los requisitos funcionales deben ser consistentes, es decir, no pueden tener definiciones contradictorias entre sí.

- ◇ RF1. Visualización de categorías: Se mostrará un listado de las categorías de videojuegos en orden ascendente, en base a su identificador en base de datos.
- ◇ RF2. Mensaje de aviso con categoría errónea: Se mostrará un mensaje de aviso cuando la categoría seleccionada no sea válida.
- ◇ RF3. Visualización de videojuegos: Se mostrará un listado con los videojuegos por categoría.
- ◇ RF4. Mensaje aviso categoría sin videojuegos: Se mostrará un mensaje de aviso cuando una categoría no tenga videojuegos relacionados con la misma.
- ◇ RF5. Extraer tweets: Se deberá poder extraer tweets de Twitter. Se descartarán siempre duplicados.
- ◇ RF6. Mensaje al extraer tweets: En caso de no encontrar tweets, se mostrará un mensaje de aviso al usuario.
- ◇ RF7. Búsqueda de usuarios-hashtag: Se buscarán tweets de los usuarios, en los *hashtags* del resto de videojuegos de la misma categoría en la que aparecieron. Se descartarán siempre duplicados.
- ◇ RF8. Nuevos *hashtags* de videojuegos: Se guardarán nuevos posibles *hashtags* en el sistema de forma automática y se descartarán siempre los duplicados.
- ◇ RF9. Mensaje al guardar tweets: Se mostrará un mensaje de éxito, cuando se haya concluido la extracción de tweets satisfactoriamente.
- ◇ RF10. Análisis de sentimiento: La herramienta realizará análisis de sentimiento de los *tweets* almacenados.
- ◇ RF11. Mensaje al analizar datos: El sistema mostrará un mensaje de aviso cuando no haya tweets que analizar.
- ◇ RF12. Almacenaje de sentimientos: La herramienta guardará el sentimiento del último *tweet* de un usuario, para un mismo videojuego. Es decir, la última opinión es la que cuenta.

- ◇ RF13. Mensaje al guardar sentimientos: Se mostrará un mensaje de éxito, cuando se hayan guardado los datos de sentimiento satisfactoriamente.
- ◇ RF14. Recomendación de Videojuegos: La herramienta mostrará los videojuegos recomendados para cada usuario, cuando encuentre una recomendación.
- ◇ RF15. Mensaje no hay recomendación de videojuegos: Cuando no se encuentre ninguna recomendación posible para los usuarios, se mostrará un mensaje de aviso que indique que no se encontraron recomendaciones.
- ◇ RF16. Recomendación orientada a empresa: el sistema devolverá un listado compuesto por usuario-videojuego-valoración, agrupado por categorías.
- ◇ RF17. Ordenación de recomendación orientada a empresa: Los usuarios promotores se ordenarán de forma descendente tomando como referencia su valoración. En cambio, los usuarios detractores, se ordenarán de manera ascendente.
- ◇ RF18. Mensaje no hay promotores/detractores: Se mostrará un mensaje de aviso, cuando no haya promotores o detractores dentro de una categoría de videojuegos.
- ◇ RF19. Cerrar: Se podrá salir/cerrar la herramienta desde el menú principal.
- ◇ RF20. Menú de herramienta: La herramienta contará con un menú principal desde el que poder realizar las diferentes acciones posibles.
- ◇ RF21. Mensaje no hay recursos API análisis: Se mostrará un mensaje indicando cuando se han agotado las peticiones diarias referentes a análisis de sentimiento.
- ◇ RF22. Mensaje no se desea realizar análisis: Se mostrará un mensaje aceptando la negativa del usuario a realizar análisis de los datos.
- ◇ RF23. Pregunta análisis de datos: Se mostrará un mensaje preguntando que se desea hacer en el menú de análisis de datos.
- ◇ RF24. Navegación: se debe poder navegar por toda la herramienta sin que el flujo se corte, en caso de algún error la herramienta debe poder recuperarse y continuar su ejecución.

3.3.2 Requisitos no funcionales

Los requisitos no funcionales, hacen referencia a propiedades del sistema, como la fiabilidad, la capacidad de almacenamiento, la usabilidad o el tiempo de respuesta. Nacen de las necesidades del usuario y podemos clasificarlos en requerimientos de producto (rendimiento, fiabilidad, portabilidad y usabilidad), organizacionales (implementación, entrega y documentación) y externos (interoperabilidad, legales y éticos)[9].

- ◆ RnF1. Fiabilidad: La herramienta deberá recuperarse, en caso de que el usuario introduzca una entrada no válida.

- ◆ RnF2. Usabilidad: La herramienta tendrá una interfaz de usuario simple, que permita al usuario entender con facilidad las acciones que debe realizar.
- ◆ RnF3. Implementación: Se desarrollará en lenguaje Java.
- ◆ RnF4. Implementación: Se usará MySQL como sistema gestor de base de datos.
- ◆ RnF5. Implementación: Se tomará como referencia el patrón Modelo-Vista-Controlador.
- ◆ RnF6. Legal: Los datos almacenados, nombres de usuarios y sus opiniones, serán gestionados siempre por una persona de la organización, en este caso, el desarrollador.
- ◆ RnF7. Rendimiento: El programa en ejecución no sobrepasará los 200Mb de uso de memoria, cuando se encuentre realizando las tareas más pesadas.
- ◆ RnF8. Legal: En caso de exponer al público, la herramienta dispondrá de una licencia de uso privada de sus servicios.
- ◆ RnF9. Usabilidad: El sistema mostrará información sobre el proceso que está llevando a cabo, en caso de que este vaya a tardar en realizarse un tiempo que exceda los 2 segundos.
- ◆ RnF10. Interoperabilidad: El sistema tardará un máximo de 2 segundos en realizar peticiones a servicios externos a la herramienta. Como en el caso de las llamadas a las diferentes APIs.
- ◆ RnF11. Implementación: Se hará uso de transacciones en todas aquellas acciones de la aplicación, que conlleven la modificación de varias tablas de la base de datos al mismo tiempo.
- ◆ RnF12. Interoperabilidad: El código desarrollado, se sincronizará con un repositorio remoto, para poder realizar gestión de versiones, dicha gestión la llevará a cabo el desarrollador.

3.4 Casos de uso

Los casos de uso, definen el comportamiento que tiene el sistema y ayudan a describir cómo se debe usar, para alcanzar un objetivo [24].

Normalmente, cada caso de uso se nombra con un verbo o una frase verbal, que expresa el objetivo que debe cumplir el propio caso. A continuación, en la figura 3.3. se muestra el diagrama de casos de uso de la herramienta desarrollada.

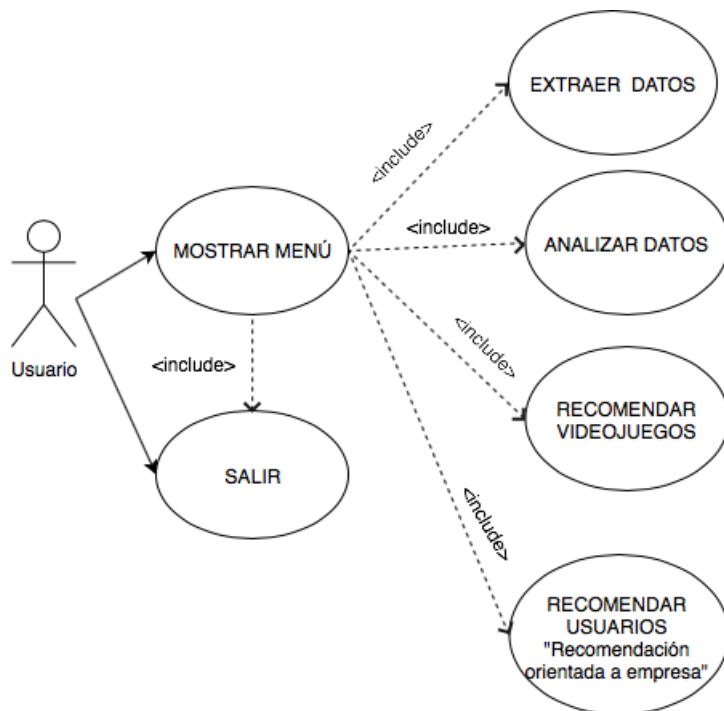


Figura 3.2 Diagrama de casos de uso del usuario de tipo empresarial

Cada caso de uso del usuario de tipo empresarial (Figura 3.3.), lo representaremos en una tabla que contiene los siguientes campos:

- Identificador y Título del caso de uso. Los identificadores siempre se formarán con la nomenclatura CUn, donde n es el número de caso de uso.
- Actor, el papel que ejerce una entidad externa al sistema, bien sea una persona o un sistema.
- Flujo de eventos, que indica los pasos a seguir para llevar a cabo la acción.
- Descripción del caso.
- Condiciones de entrada, indican el estado en el que deben estar los actores y el sistema antes de comenzar el caso de uso.
- Condiciones de salida, indican el estado en el que queda el sistema cuando finaliza el caso de uso.

CU1: Mostrar menú principal	
Actor	Usuario
Flujo de eventos	1. Ejecutar herramienta
Descripción del caso	El usuario ejecuta la herramienta y se muestra el menú principal.
Condiciones de entrada	Ejecutar herramienta
Condiciones de salida	Se muestra el menú principal

Tabla 3-1 CU1: Mostrar menú principal

CU2: Salir	
Actor	Usuario
Flujo de eventos	<ol style="list-style-type: none"> 1. Ejecutar la herramienta 2. Seleccionar “Salir”, opción 5 del menú.
Descripción del caso	El usuario sale y cierra la herramienta.
Condiciones de entrada	Ejecutar la herramienta
Condiciones de salida	Se sale de la herramienta.

Tabla 3-2 CU2: Salir

CU3: Extraer tweets	
Actor	Usuario
Flujo de eventos	<ol style="list-style-type: none"> 1. Ejecutar herramienta. 2. Seleccionar “1”, extraer datos. 3. Seleccionar una categoría. 4. Seleccionar un videojuego.
Descripción del caso	El usuario utiliza la opción que hay en la herramienta para extraer tweets.
Condiciones de entrada	Seleccionar en el menú principal la acción “Extraer datos”, opción 1 del menú.
Condiciones de salida	Se extraen tweets y se guardan en base de datos.

Tabla 3-3 CU3: Extraer tweets

CU4: Analizar datos	
Actor	Usuario
Flujo de eventos	<ol style="list-style-type: none"> 1. Ejecutar la aplicación 2. Seleccionar “2”, opción Analizar datos. 3. Seleccionar “y”, en la pregunta que aparece dentro del menú de análisis.
Descripción del caso	El usuario realiza la acción de análisis de sentimiento de los datos sin procesar.
Condiciones de entrada	Seleccionar en el menú la opción “2”.
Condiciones de salida	Se analizan los datos y se almacena el resultado en base de datos.

Tabla 3-4 CU4: Analizar datos

CU5: Recomendar videojuegos	
Actor	Usuario
Flujo de eventos	<ol style="list-style-type: none"> 1. Ejecutar la herramienta 2. Seleccionar la acción “3” del menú principal
Descripción del caso	El usuario inicia el proceso que devuelve como resultado las recomendaciones de videojuegos de los usuarios de Twitter
Condiciones de entrada	Seleccionar el menú “3” de la aplicación
Condiciones de salida	Se devuelven como resultado del proceso, las recomendaciones de videojuegos.

Tabla 3-5 CU5: Recomendar videojuegos

CU6: Recomendar usuarios (Recomendación orientada empresa)	
Actor	Usuario
Flujo de eventos	<ol style="list-style-type: none"> 1. Ejecutar la aplicación 2. Seleccionar la opción “4” del menú
Descripción del caso	El usuario decide realizar la ejecución del proceso de recomendación, para obtener los usuarios promotores y detractores de cada categoría de videojuegos.
Condiciones de entrada	Seleccionar en el menú la acción “4”
Condiciones de salida	Se devuelve un listado de los usuarios promotores/detractores por categoría

Tabla 3-6 CU5: Recomendar videojuegos

4 IMPLEMENTACIÓN DEL SISTEMA

4.1 Arquitectura software

Para el desarrollo de la herramienta se ha seguido un patrón de arquitectura software denominado Modelo-Vista-Controlador, de aquí en adelante, MVC.

Este patrón, separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

MVC se basa en la definición de tres componentes principales: el modelo, la vista y el controlador.

El modelo se compone de los datos del sistema y se encarga de su gestión, ya sea para consultarlos, manipularlos o actualizarlos. Es el encargado de enviar a la vista la información que esta le solicita, para que sea presentada al usuario. Todas estas solicitudes de datos, o su manipulación, llegan a través del controlador.

El controlador actúa de intermediario entre la vista y el modelo. Su trabajo es gestionar los eventos que provienen de la vista y realizar las solicitudes correspondientes al modelo. La vista es la encargada de representar los datos del modelo, de modo que el usuario pueda interactuar con ellos [3].

La Figura 4.1 muestra un esquema simplificado del patrón MVC del recomendador implementado en este proyecto.

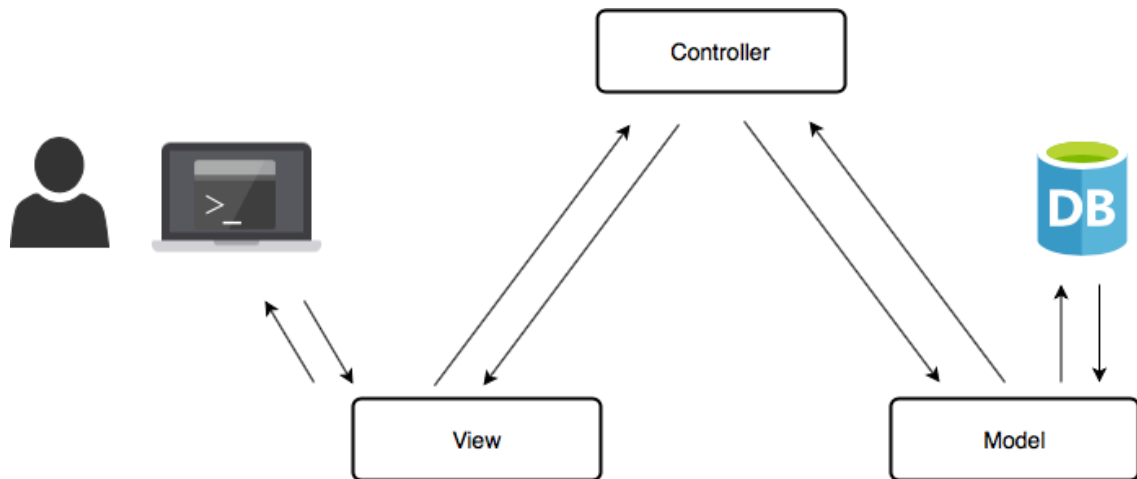


Figura 4.1 Modelo MVC de la herramienta

El conjunto de clases desarrollados para la herramienta, de acuerdo al modelo de arquitectura MVC, queda agrupado de la siguiente manera (Figura 4.2.).

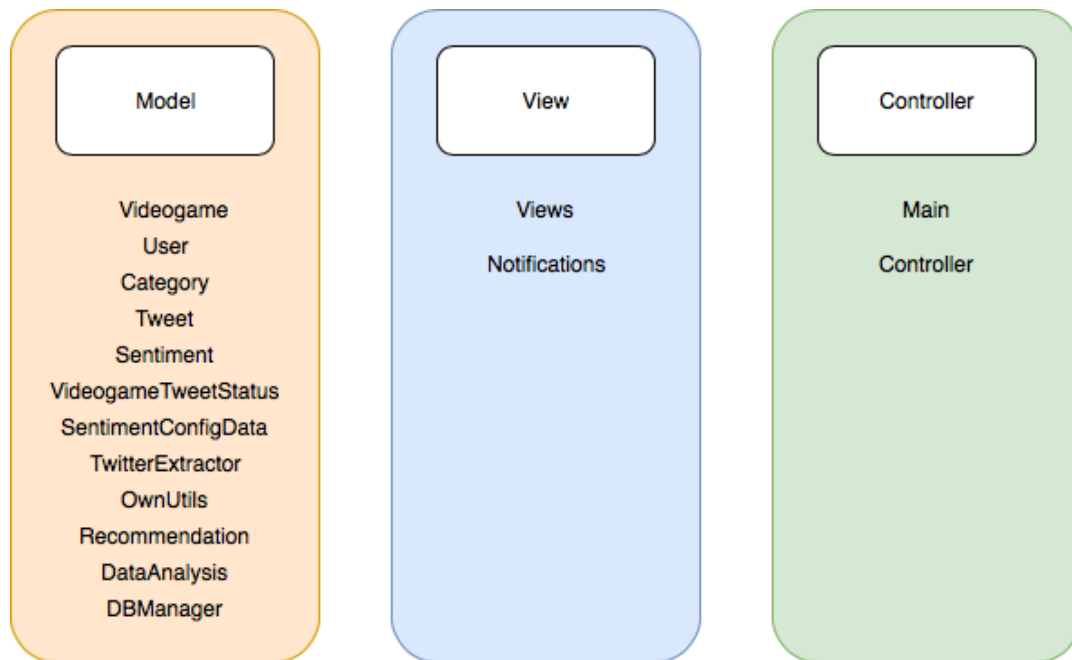


Figura 4.2 Clases agrupadas en los diferentes componentes de MVC

4.2 Clases y Métodos implementados

En este capítulo se describen los métodos de las clases implementadas en el desarrollo del recomendador de videojuegos.

La información de este apartado se representa mediante tablas con el siguiente formato.

Nombre de la clase	
Descripción de la clase	
Nombre del método	Descripción del método

- **Clases del paquete Controller**

Controller	
Clase encargada de intermediar entre el modelo y la vista.	
<code>displayMainMenu()</code>	Método que carga el menú principal de la herramienta.
<code>displayTweetExtractionMenu()</code>	Método que carga el menú de extracción de tweets.
<code>displayAnalysisDataMenu()</code>	Método que carga el menú de análisis de sentimiento de los datos.
<code>displayVideoGameRecommenderMenu()</code>	Método que carga el menú de recomendación de videojuegos.
<code>displayEnterpriseRecommenderMenu()</code>	Método que carga el menú de recomendación orientada a empresas.
<code>displayExit()</code>	Método que carga el cierre del programa.

Tabla 4-1 Clases del paquete Controller

Main	
Clase principal para el arranque de la herramienta	
<code>main()</code>	Método encargado del arranque del programa.

- **Clases del paquete View**

Notifications	
Clase encargada de representar los mensajes de estado de la herramienta.	
<code>warningMsg()</code>	Método que representa los diferentes mensajes de aviso de la herramienta.
<code>successMsg()</code>	Método que representa los diferentes mensajes de éxito de la herramienta.
<code>errorMsg()</code>	Método que representa los diferentes mensajes de error de la herramienta.
<code>displayNotification()</code>	Método que representa los mensajes de la herramienta dependiendo del código de mensaje.
<code>displayRedirMsg()</code>	Método que representa un mensaje de redirección al menú principal.

Tabla 4-2 Clases del paquete View - Notifications

Views	
Clase que representa las vistas del programa.	
<code>displayMainMenu()</code>	Método que representa el menú principal.
<code>displayHeader()</code>	Método que representa la cabecera del programa.
<code>headerText()</code>	Método que representa los textos de las cabeceras, dependiendo del código de entrada.
<code>displayTweetExtractionMenu()</code>	Método que representa el menú de extracción de tweets.

<code>displayVideoGamesFromCategory()</code>	Método que representa el listado de los videojuegos de una categoría.
<code>displayAnalyzeDataMenu()</code>	Método que representa el menú de análisis de sentimiento de datos.
<code>displayVideoGameRecommenderMenu()</code>	Método que representa el menú de recomendación de videojuegos.
<code>displayEnterpriseRecommenderMenu()</code>	Método que representa el menú de recomendación orientado a empresas.
<code>displayExit()</code>	Método que representa el cierre del programa.

Tabla 4-3 Clases del paquete View - Views

- **Clases del paquete Model**

Category	
Representación del objeto <code>category</code> , para las categorías de los videojuegos.	
No contiene métodos destacables. Sólo tiene los métodos <code>getter</code> y <code>setter</code> de los campos que componen un objeto <code>category</code> .	
DataAnalysis	
Clase en la que se implementan los métodos para realizar análisis de sentimiento	
<code>getSentiment()</code>	Método que devuelve un sentimiento sobre un texto.
<code>getSentimentValue()</code>	Método que calcula el valor de un sentimiento, dependiendo de la polaridad del mismo.

Tabla 4-4 Clases del paquete Model - Category & DataAnalysis

DBManager	
Clase encargada de la gestión de lectura y escritura desde base de datos.	
<code>connect()</code>	Método que obtiene la conexión a base de datos.
<code>close()</code>	Método que cierra la conexión a base de datos.
<code>getUserByName()</code>	Método que obtiene un usuario buscando por nombre. Si el usuario no existe, lo crea y se llama recursivamente para devolverlo.
<code>getCategories()</code>	Método que obtiene las categorías de videojuegos.
<code>getVideoGamesByCategory()</code>	Método que obtiene los videojuegos de una categoría.
<code>getVideoGameById()</code>	Método que obtiene un videojuego buscando por id.
<code>saveTweet()</code>	Método que guarda un tweet
<code>checkTweetId()</code>	Método que comprueba si existe un tweet almacenado ya en la base de datos.
<code>checkHashtag()</code>	Método que comprueba si un hashtag existe ya en la base de datos.
<code>findData2Analyze()</code>	Método que obtiene los datos que aún no han sido procesados.

findData2Analyze(remainingCalls)	Método que obtiene los datos que aún no han sido procesados, pero con un límite de resultados que marca el parámetro de entrada <i>remainingCalls</i> .
saveNewHashtag()	Método que guarda los nuevos hashtag encontrados al extraer tweets.
getSentimentConfiguration()	Método que obtiene los campos de configuración de la API de Analisis de Sentimiento.
saveSentiment()	Método que guarda un sentimiento(objeto <i>Sentiment</i>).
getAllSentiments()	Método que devuelve todos los sentimientos guardados en base de datos.
getVideoGameIdsByCategory()	Método que obtiene el id de todos los juegos de una categoría.
getPromotersCategory()	Método que obtiene usuarios promotores de una categoría de videojuegos.
getDetractorsCategory()	Método que obtiene usuarios detractores de una categoría de videojuegos.
updateSentimentConfigData()	Método encargado de revisar y actualizar los datos de la configuración de la API de Análisis de Sentimiento.
getUserNameByUserId()	Método que devuelve el nombre de un usuario.
getAllSentimentDataUsers()	Método que obtiene un listado de todos los usuarios que aparecen en los sentimientos.

Tabla 4-5 Clases del paquete Model - DBManager

OwnUtils	
Clase de utilidades	
sleepMs()	Método para dormir el programa durante X milisegundos.
isNumber()	Método que comprueba que un String se pueda parsear a Long.
pressAnyKey()	Método que para el programa hasta que se presiona una tecla.
replaceCharactersToMysql()	Método para reemplazar determinados caracteres de un texto, que provocan un error al intentar introducirlos en base de datos.

Tabla 4-6 Clases del paquete Model - OwnUtils

Recommendation	
Clase que representa al objeto de una recomendación(<i>Recommendation</i>).	
recCsvWriter()	Método que genera un fichero CSV de recomendaciones de videojuegos.
executeRecommender()	Método que lanza el recomendador.

Tabla 4-7 Clases del paquete Model - Recommendation

Sentiment	
Clase que representa el objeto sentimiento(<i>Sentiment</i>).	
saveSentiments ()	Método que guarda los sentimientos(objetos <i>Sentiment</i>).
sentimentCsvWriter ()	Método que genera un fichero CSV de los análisis de sentimiento.

Tabla 4-8 Clases del paquete Model - Sentiment

SentimentConfigdata	
Clase que representa el objeto <i>SentimentConfigData</i> que almacena los datos de configuración de la API de Análisis de Sentimiento.	
checkAndUpdateSentimentConfig ()	Método que comprueba y actualiza el estado de los campos de configuración de la API de Análisis de Sentimiento.

Tabla 4-9 Clases del paquete Model - SentimentConfigdata

Tweet	
Clase que representa el objeto <i>tweet</i> .	
turnVideogameTweets2Tweets ()	Método que convierte objetos <i>VideogameTweetStatus</i> en objetos de tipo <i>Tweet</i> .
saveTweets ()	Método que guarda los tweets extraídos en la base de datos.

Tabla 4-10 Clases del paquete Model - Tweet

TwitterExtractor	
Clase que contiene los métodos relativos a la extracción de tweets.	
twitterConfig ()	Método que obtiene un objeto <i>Twitter</i> de configuración, para poder realizar llamadas a la API Search de Twitter.
extract ()	Método que obtiene tweets de un hashtag determinado.
findTweetByHashtagAndUser ()	Método que busca tweets de un determinado hashtag y usuario.

Tabla 4-11 Clases del paquete Model - TwitterExtractor

User	
Representación del objeto <i>User</i> , los usuarios de Twitter.	
No contiene métodos destacables. Sólo tiene los métodos <i>getter</i> y <i>setter</i> de los campos que componen un objeto <i>User</i>	

Tabla 4-12 Clases del paquete Model - User

Videogame	
Clase que representa el objeto Videogame, videojuego.	
saveNewHashtags ()	Método que obtiene nuevos posibles hashtag de videojuegos y los guarda en base de datos.

Tabla 4-13 Clases del paquete Model - Videogame

VideogameTweetStatus	
Clase que representa el objeto VideogameTweetStatus. Se trata de un objeto intermedio que se necesita para poder convertir los objetos Status, que provienen de la API Search de Twitter, en objetos VideogameTweet.	
turnStatuses2VideogameTweets ()	Método que convierte objetos Status en objetos VideogameTweet.

Tabla 4-14 Clases del paquete Model - VideogameTweetStatus

4.3 Funcionalidad del sistema

En esta sección, se representan los diagramas de flujo de la herramienta. Estos diagramas ayudan a comprender el comportamiento que tiene la aplicación en cada uno de sus módulos.

El diagrama de la Figura 4.3 representa el modo en que un usuario puede navegar a través del menú principal. Cómo se puede observar, se trata de un flujo sencillo en el que basta con seleccionar una de las cinco acciones posibles. En caso de que la acción no exista, se pide de nuevo al usuario que elija una opción dentro de las definidas. Por otro lado, si el usuario decide elegir la opción “Salir”, se produce el cierre completo de la herramienta.

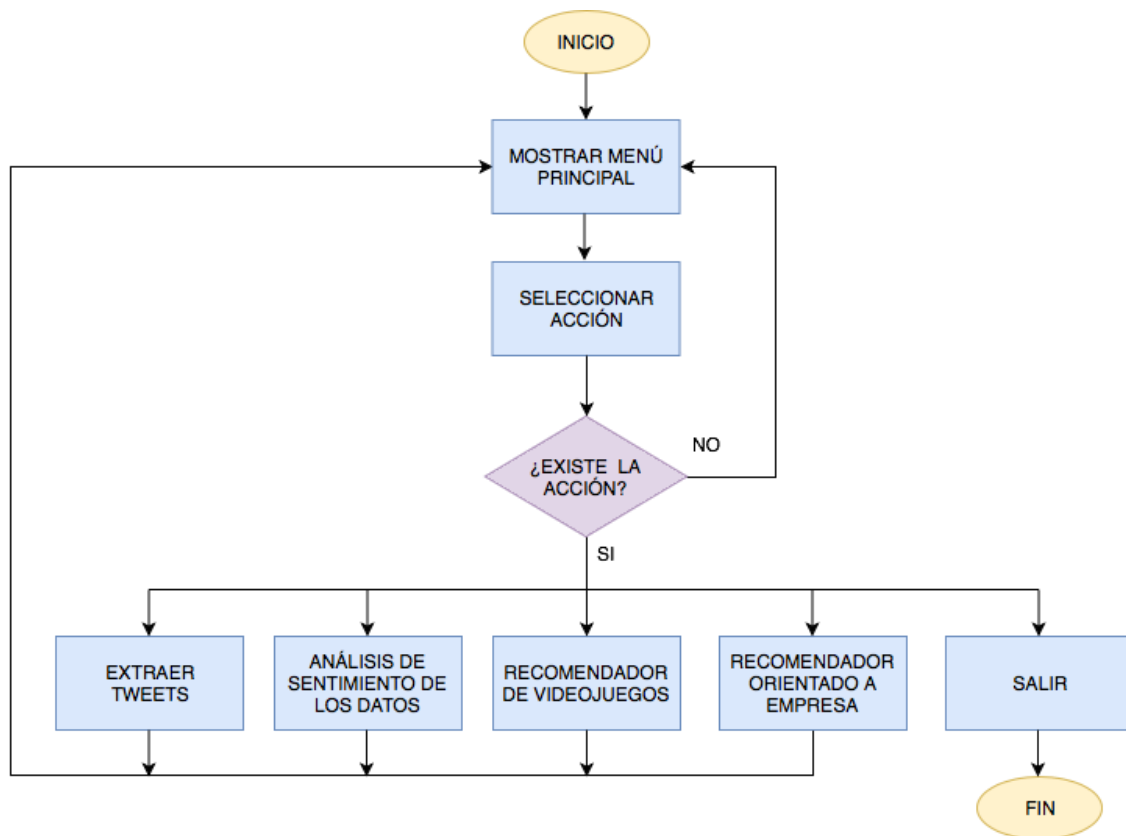


Figura 4.3 Diagrama de flujo del Menú principal

El diagrama que tenemos más abajo (Figura 4.4.), ayuda a entender el flujo que sigue la aplicación a la hora en que un usuario decide realizar la extracción de tweets.

Lo primero que debe hacer el usuario es seleccionar entre las categorías de videojuegos existentes. Si la selección no es válida, se devuelve al usuario al mismo punto en el que empezó, es decir, la elección de categoría. Y si la categoría no tiene videojuegos, también devolvemos al usuario al punto de partida.

El siguiente paso, una vez se ha escogido la categoría, es la selección de un videojuego dentro de dicha categoría. De nuevo, si la elección no es válida, se lleva al usuario al inicio del menú de extracción de tweets.

Cuando ya tenemos categoría y videojuego, la herramienta comienza la búsqueda de tweets por el hashtag del videojuego.

Por un lado, si no se encuentran tweets, se retorna al usuario al menú principal. Por otro lado, si se encuentran tweets se comienza una segunda búsqueda de tweets, pero esta vez buscamos a los usuarios de los tweets extraídos, en el resto de videojuegos de la categoría. Con esto se intenta encontrar relaciones entre las personas que opinan sobre los mismos videojuegos.

Tras la segunda búsqueda, tanto si encontramos nuevos tweets, cómo si no, procedemos a analizar si los tweets encontrados, tienen nuevos posibles hashtag con los que alimentar de forma automática nuestra base de datos de los videojuegos de una misma categoría. Si contienen hashtag, se descartan los duplicados y se guardan sólo los nuevos.

Una vez acabada la tarea anterior, se procede al guardado de los tweets, siempre descartando los duplicados. Finalmente, acaba el proceso de extracción, indicando al usuario cuantos tweets se encontraron y guardaron, y se devuelve al usuario al menú principal de la herramienta.

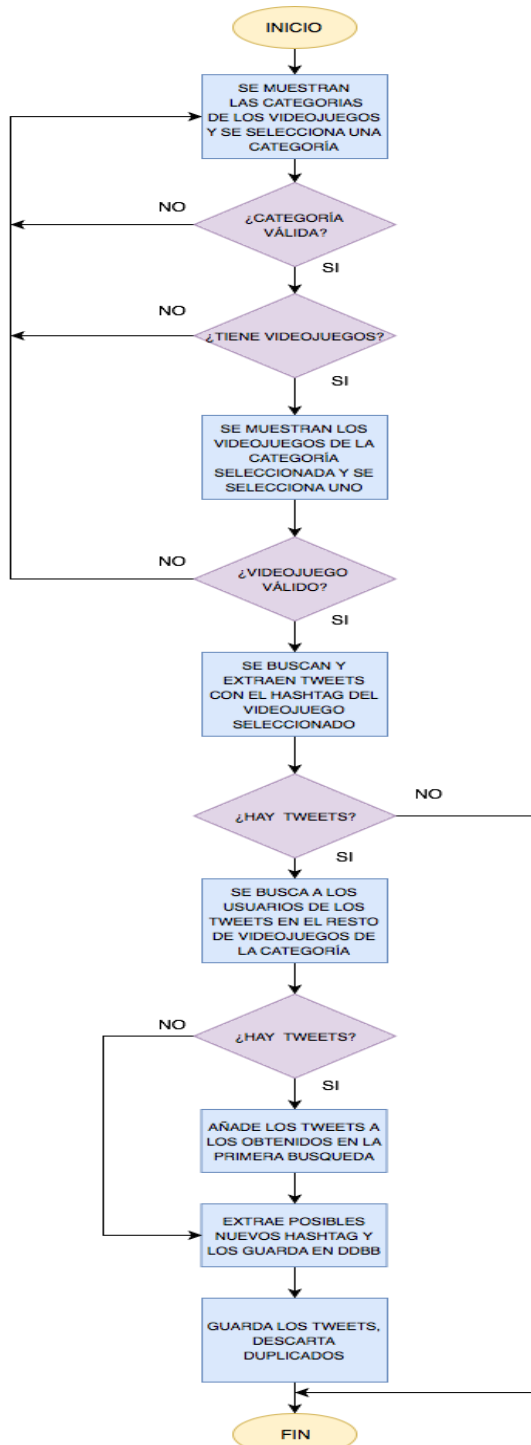


Figura 4.4 Diagrama de flujo de la extracción de tweets

La Figura 4.5. muestra cómo se realiza el proceso de análisis de sentimiento dentro de la herramienta.

Lo primero que se comprueba es si hay datos que analizar. Si no los hay, se redirige al usuario al menú principal de la aplicación. En caso de que los haya, se pregunta al usuario si desea procesarlos o no.

Si la respuesta es no, se lleva al usuario al menú principal.

Si la respuesta es que si, se ejecuta el proceso.

1. Obtenemos todos los datos almacenados sin procesar.
2. Realizamos las peticiones al servicio de análisis de sentimiento.
3. Almacenamos en base de datos el listado completo de todos los datos procesados.

Una vez el proceso termina, se redirige al usuario al menú principal.

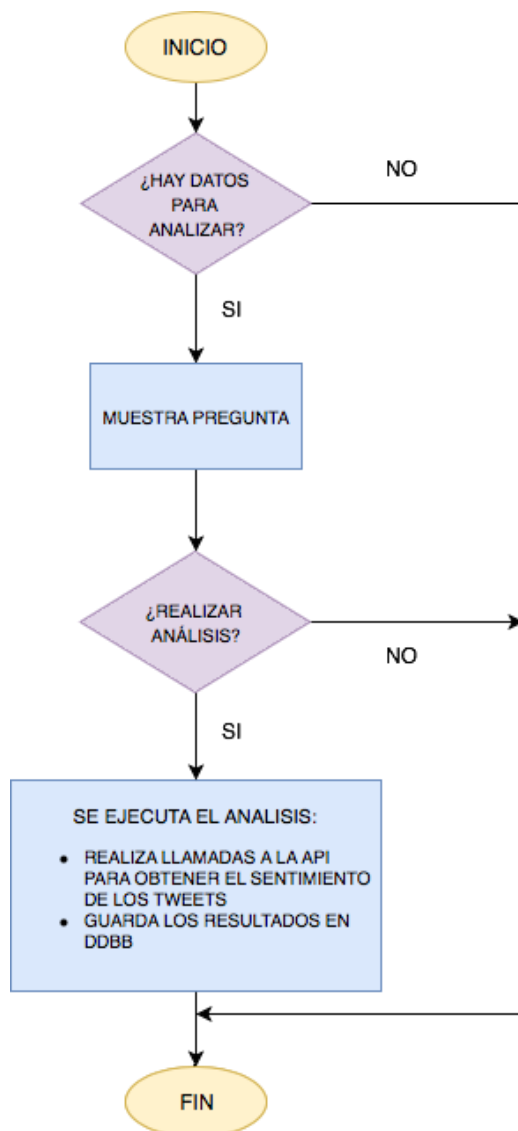


Figura 4.5 Diagrama de flujo de análisis de sentimiento

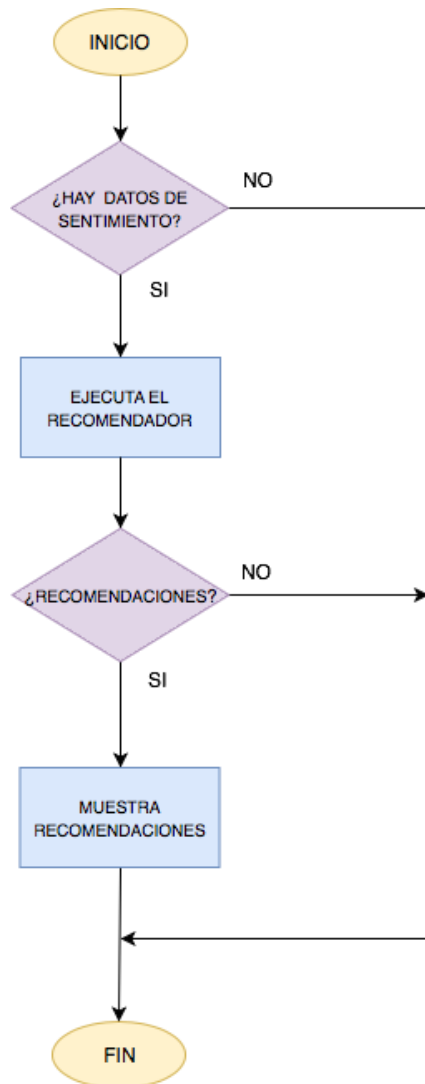


Figura 4.6 Diagrama de flujo de recomendación de videojuegos

El diagrama anterior (Figura 4.6.) indica el funcionamiento del menú de recomendación de videojuegos.

Lo primero que se hace es comprobar que hay datos que se puedan analizar para realizar recomendaciones. Si no los hay, se devuelve al usuario al menú principal. En caso de haber datos, se ejecuta el recomendador.

Si encuentra videojuegos que recomendar, muestra las recomendaciones para cada usuario. De lo contrario, acaba el proceso de recomendación.

Cuando acaba, el usuario tiene la posibilidad de quedarse mirando los datos, o de ir al menú principal.

La Figura 4.7. representa el flujo que se sigue dentro del menú de recomendación orientada a empresa.

Se trata de un menú muy sencillo, donde se ejecuta el recomendador orientado a empresa y se muestran las recomendaciones para cada categoría, que se encuentren.

Una vez finalizado el proceso, el usuario puede decidir cuándo ir al menú principal.

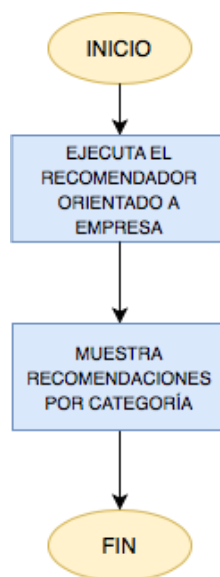


Figura 4.7 Diagrama de flujo de recomendación orientada a empresa

4.4 Interfaz del usuario

La interfaz de usuario es el componente de la aplicación, mediante el que un usuario interactúa. En el recomendador, la interfaz se presenta a través de terminal y el usuario se comunica con esta por medio de órdenes desde el teclado.

Lo primero que se encuentra el usuario es un menú principal, desde el que puede seleccionar qué acción quiere realizar (Figura 4.8).

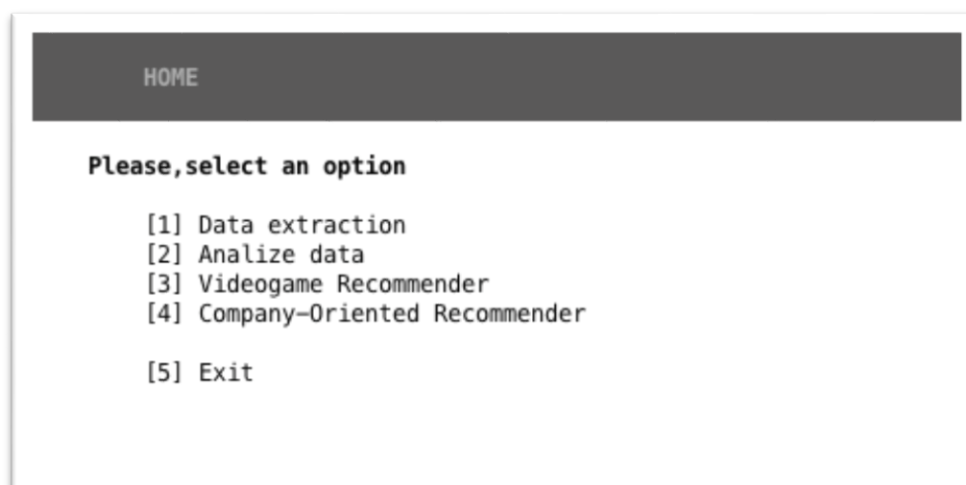


Figura 4.8 Interfaz que representa la pantalla del menú principal del recomendador

Tomando el menú principal como punto de referencia, el usuario podrá realizar cualquiera de las acciones que aparecen en la figura anterior, extraer datos de Twitter, analizar los datos extraídos, mirar las recomendaciones de videojuegos por usuario y ver las recomendaciones orientadas a empresa.

A continuación(Figura 4.9.), se puede ver el comportamiento de la herramienta durante la extracción de datos de Twitter. Dónde el usuario, selecciona tanto la categoría como el videojuego del que se buscarán “tweets”.

La herramienta realiza una búsqueda de “tweets” y al acabar, guarda los datos extraídos.

DATA EXTRACTION

Please,select a videogame category using the number within []

[2]Shooter
[3]Race
[4]Sport

3

Videogames from category selected

Please,select a videogame using the number within []

[14]F12017
[15]F12016
[16]NFSPayback
[17]NeedforSpeed
[18]GTSport
[19]GranTurismo7
[20]Forza7
[21]ProjectCars2

21

Videogame selected: ProjectCars2
10 tweets found
Searching users by hashtag

#F12017 searching 100%	<div></div>	10/10 (0:00:25 / 0:00:00)
#F12016 searching 100%	<div></div>	10/10 (0:00:24 / 0:00:00)
#NFSPayback searching 100%	<div></div>	10/10 (0:00:24 / 0:00:00)
#NeedforSpeed searching 100%	<div></div>	10/10 (0:00:25 / 0:00:00)
#GTSport searching 100%	<div></div>	10/10 (0:00:24 / 0:00:00)
#GranTurismo7 searching 100%	<div></div>	10/10 (0:00:24 / 0:00:00)
#Forza7 searching 100%	<div></div>	10/10 (0:00:24 / 0:00:00)

Tweets saved successfully: 10

Figura 4.9 Interfaz que representa la pantalla para extraer tweets de videojuegos

En el caso de la pantalla de análisis, aparece una pregunta, en la que se indica si se desea realizar el análisis de los datos. En caso afirmativo, el sistema procesa los datos y al terminar, los guarda. Si, por el contrario, se decide no realizar el análisis, Se vuelve al menú principal.

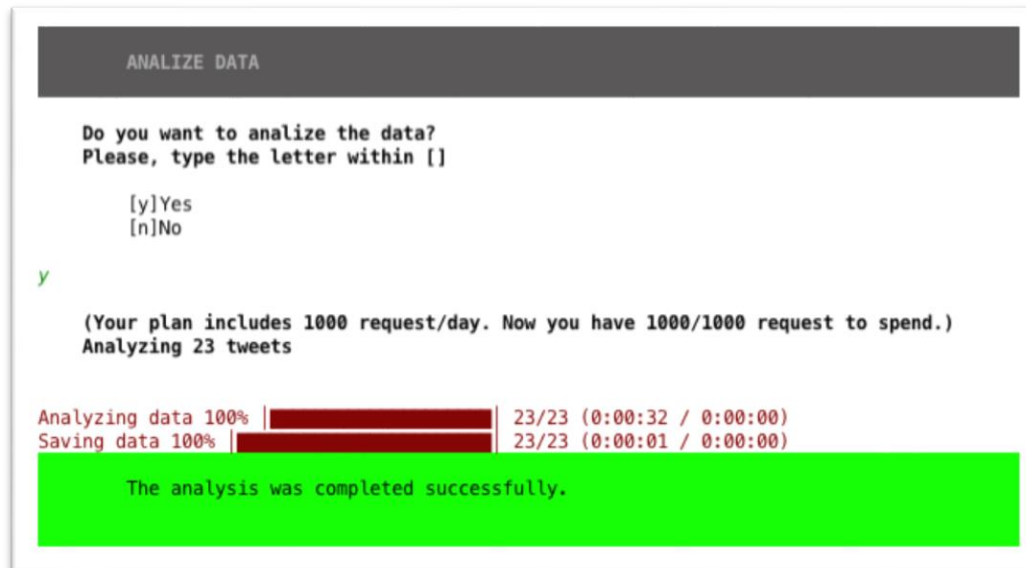


Figura 4.10 Interfaz que representa la pantalla del menú para análisis de sentimiento de los datos

La siguiente imagen (Figura 4.11), representa las recomendaciones de los videojuegos que ha devuelto la herramienta, para una serie de usuarios.

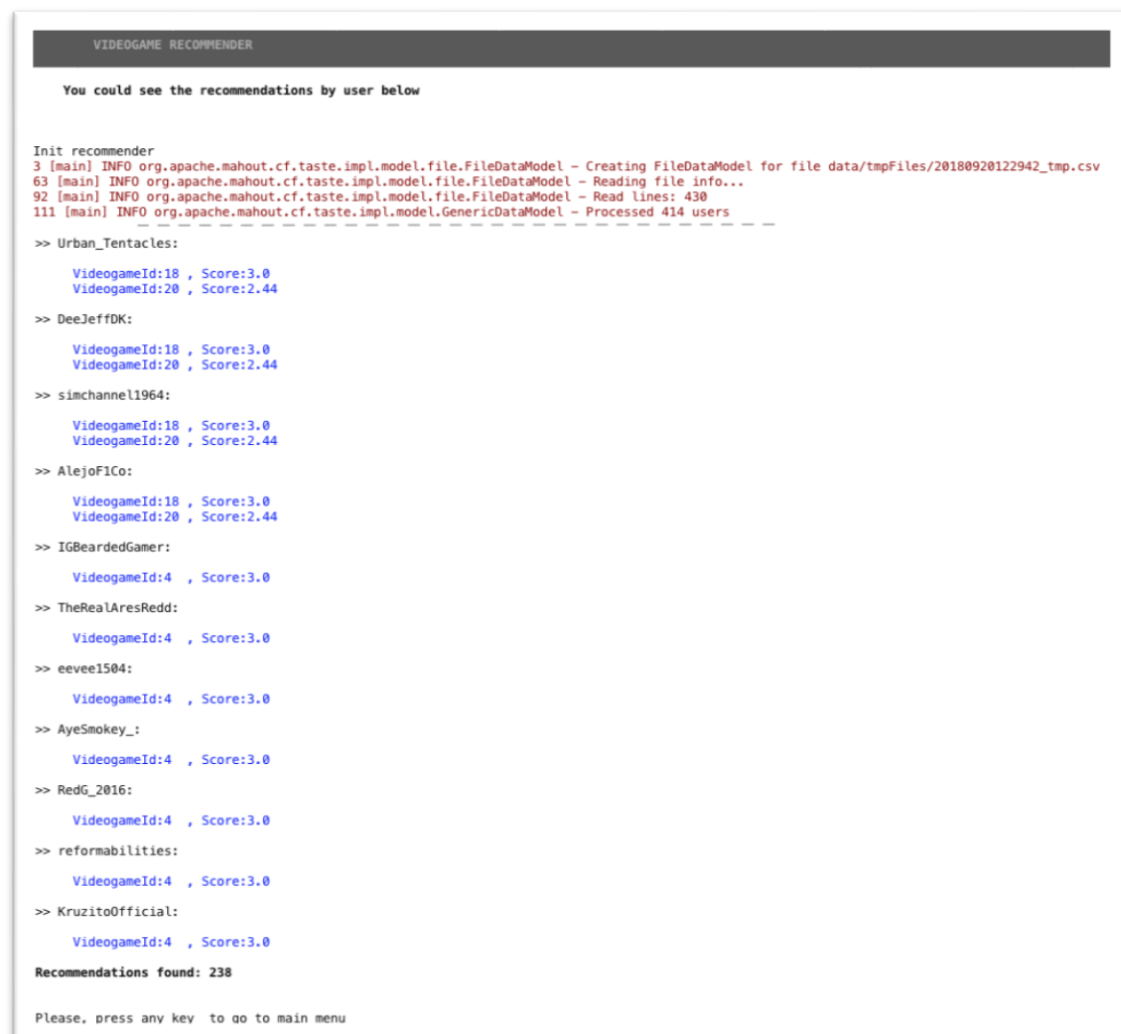


Figura 4.11 Interfaz que representa la pantalla de recomendación de videojuegos

La Figura 4.12. Pone de manifiesto, la representación que se hace por consola, cuando se pide ejecutar el proceso de recomendación orientado a empresa, que devuelve usuarios promotores/detractores por categoría de videojuego.

```

COMPANY-ORIENTED RECOMMENDER

You could see the company-oriented recommendations by category below

Shooter:
Promoters:
  UserId:23      VideogameId:1      Score:4.99
  UserId:388     VideogameId:7      Score:4.99
  UserId:25      VideogameId:1      Score:4.97
  UserId:115     VideogameId:4      Score:4.97
  UserId:162     VideogameId:7      Score:4.95
  UserId:42      VideogameId:37     Score:4.94
  UserId:108     VideogameId:4      Score:4.91
  UserId:374     VideogameId:7      Score:4.84
  UserId:417     VideogameId:1      Score:4.84
  UserId:155     VideogameId:7      Score:4.83
Detractors:
  UserId:54      VideogameId:2      Score:1.06
  UserId:167     VideogameId:7      Score:1.5
  UserId:51      VideogameId:37     Score:1.72
  UserId:136     VideogameId:5      Score:1.74
  UserId:24      VideogameId:1      Score:1.82
  UserId:2       VideogameId:1      Score:2.14

Race:
Promoters:
  UserId:233     VideogameId:18     Score:4.98
  UserId:243     VideogameId:18     Score:4.74
  UserId:240     VideogameId:18     Score:4.74
  UserId:84      VideogameId:14     Score:4.72
  UserId:100     VideogameId:16     Score:4.52

```

Figura 4.12 Interfaz que representa la pantalla de recomendación orientada a empresa.

5 PRUEBAS

En este apartado de la documentación, se realizarán una serie de pruebas sobre la herramienta desarrollada, para comprobar el funcionamiento de las diferentes partes y concretar si se cumplen los requisitos funcionales expuestos en el apartado 3.3.1. Después se determinará si el resultado de cada una de ellas es satisfactorio o no.

Como entorno de pruebas, se ha utilizado el entorno de desarrollo IntelliJ, sobre un ordenador Apple Macbook Pro 13', donde se tiene ejecutando el programa de recomendación implementado en este TFG.

5.1 Pruebas de interfaz

Las pruebas de interfaz nos permitirán identificar si la usabilidad de la herramienta y el flujo entre las diferentes pantallas es el adecuado.

Estas pruebas, seguirán la siguiente nomenclatura: T-UIn (Test de Interfaz de Usuario), donde n refleja el número de la prueba.

T-Uin: Título de la prueba	
Descripción de la prueba	
Pasos	Resultado OK / KO
Requisito	

Tabla 5-1 T-Uin: Título de la prueba

T-UI1:Mostrar menú principal	
Al arrancar la aplicación, comprobar que lo primero que se muestra es el menú principal de la herramienta.	
Pasos	Resultado OK
<ol style="list-style-type: none"> 1. Ejecutamos la aplicación 2. Comprobamos que se muestran las diferentes opciones del menú principal. <ol style="list-style-type: none"> a. Extraer datos b. Analizar datos c. Recomendar videojuegos d. Recomendación orientada a empresas e. Salir 	
RF20	

Tabla 5-2 T-UI1:Mostrar menú principal

T-UI2:Navegar por el menú principal	
Cuando nos situamos en el menú principal, probamos que se puede navegar a los diferentes menús de la herramienta.	
Pasos	Resultado OK
<ol style="list-style-type: none"> 1. Ejecutamos la aplicación. 2. Seleccionamos cualquiera de las 5 opciones disponibles en el menú principal. 3. Comprobamos que navega al menú de la opción seleccionada. 	
RF20	

Tabla 5-3 T-UI2:Navegar por el menú principal

T-UI3:Mostrar categorías	
Si navegamos a la primera funcionalidad del menú principal, comprobar que lo primero que vemos es un listado de las categorías de videojuegos disponibles.	
Pasos	Resultado OK
<ol style="list-style-type: none"> 1. Ejecutamos la aplicación. 2. Seleccionamos 1 en el menú principal. 3. Comprobamos que aparecen las categorías de los videojuegos. 	
RF1	

Tabla 5-4 T-UI3:Mostrar categorías

T-UI4:Mostrar videojuegos por categoría

Dentro de la acción 1 del menú principal, una vez hemos seleccionado una categoría, comprobar que se muestra un listado de los videojuegos que se encuentran definidos en esa categoría.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación.2. Seleccionamos 1.3. Seleccionamos 3.4. Comprobamos que devuelve los videojuegos de la categoría 3.	OK
RF3	

Tabla 5-5 T-UI4:Mostrar videojuegos por categoría

T-UI5:Mostrar mensaje aviso si no hay videojuegos

Dentro de la acción 1 del menú principal, una vez hemos seleccionado una categoría, comprobar que se muestra un mensaje de aviso, indicando que no hay videojuegos, cuando una categoría no tiene videojuegos definidos.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación.2. Seleccionamos 1.3. Seleccionamos 4.4. Comprobamos que muestra un mensaje indicando que no hay videojuegos para la categoría seleccionada.	OK
RF4	

Tabla 5-6 T-UI5:Mostrar mensaje aviso si no hay videojuegos

T-UI6:Porcentaje de búsqueda de datos

Dentro de la acción 1 del menú principal, una vez hemos seleccionado una categoría y un videojuego, comprobar que aparecen barras de carga que indican el porcentaje de búsqueda que lleva de cada videojuego.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación.2. Seleccionamos 1.3. Seleccionamos 2.4. Seleccionamos 1.5. Comprobamos que mientras realiza la búsqueda de los videojuegos, muestra barras de carga que indican el porcentaje de cada uno.	OK
RnF9	

Tabla 5-7 T-UI6:Porcentaje de búsqueda de datos

T-UI7:Navegación al acabar una extracción de datos

Comprobar que cuando se acaba la extracción de datos satisfactoriamente, se transiciona hacia el menú principal

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación.	OK

2. Seleccionamos “1”.
3. Seleccionamos “2”.
4. Seleccionamos “1”.
5. Cuando acaba el proceso, comprobamos que nos redirige al menú principal.

RF24

Tabla 5-8 T-UI7:Navegación al acabar una extracción de datos

T-UI8:Mensaje para analizar datos

Si navegamos hacia el menú análisis de sentimiento de datos, comprobar que se nos pregunta si deseamos analizar datos. Esto sólo ocurre en el caso de que haya datos aún sin procesar.

Pasos

1. Ejecutamos la aplicación.
2. Seleccionamos “2”.
3. Se muestra pregunta sobre lo que queremos hacer. Si analizar los datos o no.

Resultado

OK

RF23

Tabla 5-9 T-UI8:Mensaje para analizar datos

T-UI9:Decisión negativa al analizar datos

Si navegamos hacia el menú análisis de sentimiento de datos, comprobar que si contestamos que no queremos analizar datos, se transiciona directamente al menú principal.

Pasos

1. Ejecutamos la aplicación.
2. Seleccionamos “2”.
3. Seleccionamos “n”.
4. Redirecciona al menú principal.

Resultado

OK

RF22

Tabla 5-10 T-UI9:Decisión negativa al analizar datos

T-UI10:Mensaje de ausencia de datos para analizar

Si navegamos hacia el menú análisis de sentimiento de datos, cuando no hay datos , se muestra un mensaje de aviso y se transiciona directamente al menú principal.

Pasos

1. Ejecutamos la aplicación.
2. Seleccionamos “2”.
3. Se muestra mensaje de aviso y nos redirecciona al menú principal.

Resultado

OK

RF11

Tabla 5-11 T-UI10:Mensaje de ausencia de datos para analizar

T-UI11:Mensaje aviso por agotar recursos

Comprobar que cuando hemos agotado las peticiones por día de la API de análisis, se nos devuelve un mensaje de aviso.

Pasos	Resultado
<ol style="list-style-type: none"> 1. Ejecutamos la aplicación. 2. Seleccionamos “2”. 3. Se muestra mensaje de aviso y nos redirecciona al menú principal. 	OK
RF21	

Tabla 5-12 T-UI11:Mensaje aviso por agotar recursos

T-UI12:Mensaje de ausencia de recomendaciones	
La aplicación muestra un mensaje indicando que no hay recomendaciones, cuando no se encuentra.	
Pasos <ol style="list-style-type: none">1. Ejecutamos la aplicación.2. Seleccionamos “3”.3. Se muestra mensaje indicando que no hay recomendaciones.	Resultado OK
RF15	

Tabla 5-13 T-UI12:Mensaje de ausencia de recomendaciones

T-UI13:Listado de recomendaciones de videojuegos	
La herramienta muestra un listado de los usuarios y los videojuegos recomendados para estos, cuando encuentra recomendaciones.	
Pasos <ol style="list-style-type: none">1. Ejecutamos la aplicación.2. Seleccionamos “3”.3. Se muestra el listado de videojuegos recomendados por usuario.	Resultado OK
RF14	

Tabla 5-14 T-UI13:Listado de recomendaciones de videojuegos

T-UI14:Mensaje de aviso ausencia de promotores/detractores	
Dentro del menú de Recomendación orientada a empresa, comprobar que se devuelven los mensajes que avisan al usuario sobre la ausencia de promotores/detractores dentro de una categoría de videojuegos, cuando no los hay.	
Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación.2. Seleccionamos “4”.3. Se muestra, dentro de cada categoría, un mensaje indicando que no hay promotores / detractores.	OK
RF18	

Tabla 5-15 T-UI14:Mensaje de aviso ausencia de promotores/detractores

T-UI15:Listado de promotores/detractores	
Dentro del menú de Recomendación orientada a empresa, comprobar que se devuelve el listado de promotores/ detractores en el orden correcto. Promotores en orden descendente y detractores en orden ascendente.	
Pasos	Resultado

<ol style="list-style-type: none"> 1. Ejecutamos la aplicación. 2. Seleccionamos “4”. 3. Se muestra el listado de promotores/detractores de cada categoría, en el orden correcto. 	OK
RF16	

Tabla 5-16 T-UI15:Listado de promotores/detractores

5.2 Pruebas funcionales

Las pruebas funcionales nos ayudaran a justificar que la herramienta funciona bien a la hora de llevar a cabo las acciones desarrolladas.

Estas pruebas, seguirán la siguiente nomenclatura: T-Fn (Test de Funcionalidad), donde n refleja el número de la prueba.

T-Fn: Título de la prueba	
Descripción de la prueba	
Pasos	Resultado OK / KO

T-F1:Extracción de tweets	
La herramienta extrae tweets con el hashtag del videojuego indicado	
Pasos <ol style="list-style-type: none"> 1. Ejecutamos la aplicación, 2. Seleccionamos “1”. 3. Seleccionamos una categoría. 4. Seleccionamos un videojuego. 5. Se extraen tweets del videojuego seleccionado. 	Resultado OK
RF5	

Tabla 5-17 T-F1:Extracción de tweets

T-F2:Matching de usuarios en otros videojuegos	
Cuando extraemos tweets, la herramienta busca, a los usuarios del videojuego seleccionado, en el resto de videojuegos de la categoría. Si han escrito un tweet en alguno de los otros videojuegos, los guarda en la base de datos.	
Pasos <ol style="list-style-type: none"> 1. Ejecutamos la aplicación, 2. Seleccionamos “1”. 3. Seleccionamos una categoría. 4. Seleccionamos un videojuego. 5. Se extraen tweets del videojuego seleccionado. 6. Se hace match de los usuarios en otros videojuegos, si escribieron un tweet en alguno de ellos. 	Resultado OK
RF7	

Tabla 5-18 T-F2:Matching de usuarios en otros videojuegos

T-F3:Nuevos videojuegos (retroalimentación automática)

Al extraer tweets, se hace una búsqueda de nuevos posibles hashtag(videojuegos) dentro de los tweets, para añadirlos posteriormente a la categoría correspondiente.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación,2. Seleccionamos “1”.3. Seleccionamos una categoría.4. Seleccionamos un videojuego.5. Se extraen tweets del videojuego seleccionado.6. Se buscan hashtag nuevos en los tweets y se guardan.	OK

RF8

Tabla 5-19 T-F3:Nuevos videojuegos (retroalimentación automática)

T-F4:Descarte de tweets duplicados

Al extraer tweets, antes de guardarlos, se comprueba que no estén ya almacenados en la base de datos, si están, se descartan.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación,2. Seleccionamos “1”.3. Seleccionamos una categoría.4. Seleccionamos un videojuego.5. Se extraen tweets del videojuego seleccionado.6. Antes de guardarlos se comprueba que no estén ya almacenados, en caso de coincidencia, se descarta el tweet duplicado.	OK

RF7

Tabla 5-20 T-F4:Descarte de tweets duplicados

T-F5:Descarte de hashtag duplicados

Al encontrar nuevos posibles hashtag, se comprueba que no coincidan con hashtag almacenados previamente. Si ya los tenemos almacenados, los descartamos.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación,2. Seleccionamos “1”.3. Seleccionamos una categoría.4. Seleccionamos un videojuego.5. Se extraen tweets del videojuego seleccionado.6. Antes de guardar los nuevos <i>hashtags</i> que se encuentren, se comprueba que no estén ya almacenados. En caso de coincidencia, se descarta el hashtag duplicado.	OK

RF8

Tabla 5-21 T-F5:Descarte de hashtag duplicados

T-F6:Restricción análisis de datos

Para el análisis de sentimiento de los datos, se utiliza una API con restricciones sobre las peticiones que se pueden realizar. Comprobar que no se nos permite analizar datos si hemos agotado los recursos del día.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación,2. Seleccionamos “1”.3. Seleccionamos “y”.4. La herramienta no nos permite analizar datos porque excedimos los recursos de la API para el día.	OK
RF10	

Tabla 5-22 T-F6:Restricción análisis de datos

T-F7:Análisis de datos

Comprobar la herramienta ejecuta bien el proceso de análisis de sentimiento de los datos.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación,2. Seleccionamos “1”.3. Seleccionamos “y”.4. La herramienta ejecuta el análisis de los datos sin procesar.	OK
RF10	

Tabla 5-23 T-F7:Análisis de datos

T-F8:Guardar análisis de sentimiento

Al analizar los tweets, que no hayan sido procesados aún, se guardan correctamente los resultados en la base de datos.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación,2. Seleccionamos “1”.3. Seleccionamos “y”.4. Al finalizar el análisis, los resultados se guardan correctamente.	OK
RF12	

Tabla 5-24 T-F8:Guardar análisis de sentimiento

T-F9:Recomendar videojuegos

La herramienta devuelve videojuegos a cada usuario, en caso de obtener al menos una recomendación para el usuario.

Pasos	Resultado
<ol style="list-style-type: none">1. Ejecutamos la aplicación,2. Seleccionamos “3”.3. Se ejecuta el recomendador y devuelve recomendaciones de videojuegos, a cada usuario, en caso de encontrarlas.	OK
RF14	

Tabla 5-25 T-F9:Recomendar videojuegos

T-F10: Recomendaciones orientadas a empresa

La aplicación devuelve recomendaciones de usuarios para cada categoría, en caso de encontrarlos.

Pasos

1. Ejecutamos la aplicación,
2. Seleccionamos “4”.
3. Se ejecuta el recomendador y devuelve recomendaciones de usuarios, de cada categoría, en caso de encontrarlas.

Resultado

OK

RF16

Tabla 5-26 T-F10: Recomendaciones orientadas a empresa

Además, se han realizado pruebas sobre datos reales extraídos con la herramienta, comparando los diferentes métodos de cálculo de similitud de los usuarios, que proporciona Mahout, en base a la variación del *threshold neighborhood*, para determinar que opción devuelve mejores resultados.

Los resultados se han representado en la gráfica que se muestra a continuación.

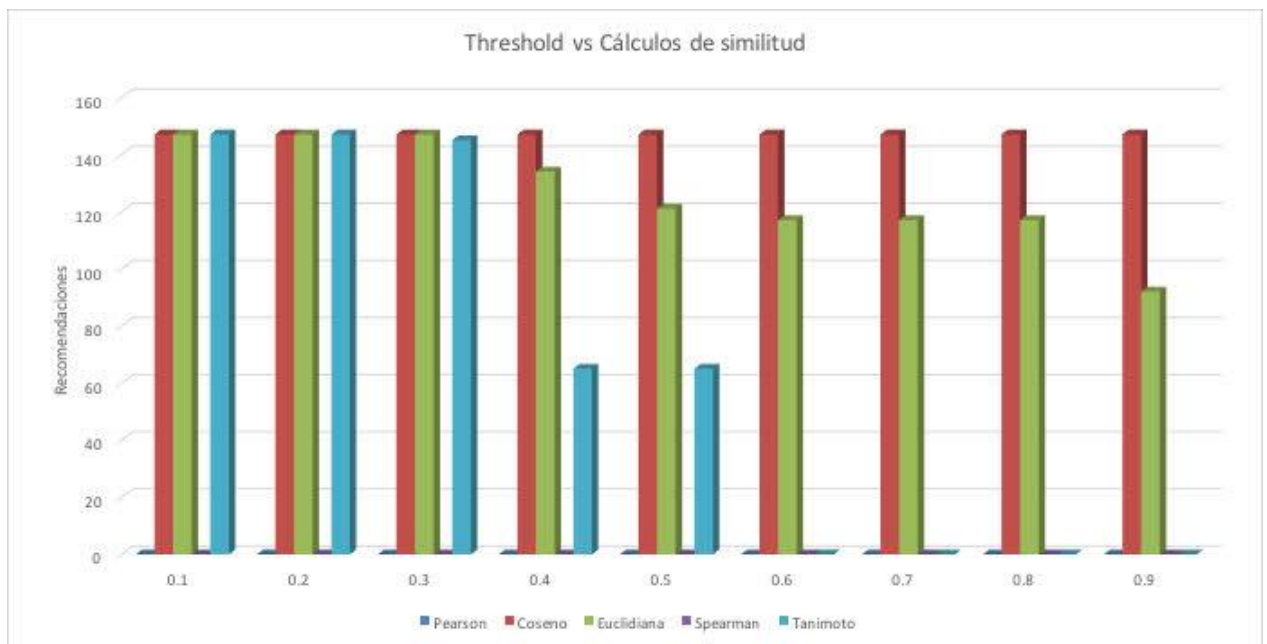


Figura 5.1 Gráfica de evaluación de recomendaciones en base a threshold vs cálculo de similitud

Como se puede observar, el cálculo de similitud mediante coseno, devuelve siempre recomendaciones, pero esto no quiere decir que sea la mejor opción, ya que es invariante con respecto al *threshold* aplicado, por lo que indica que esta devolviendo recomendaciones que pueden no ser muy precisas.

En cambio, si nos fijamos en la progresión que sigue la similitud por distancia euclidiana, tiene mayor coherencia, ya que a medida que elevamos el *threshold*, vamos perdiendo recomendaciones. Esto nos indica que las que se devuelven serán más precisas, ya que los usuarios serán más afines.

6 PLANIFICACIÓN Y PRESUPUESTO

6.1 Planificación

En este apartado, se describe la progresión ha seguido el proyecto hasta su finalización. El proyecto se ha llevado a cabo en cuatro fases. Una primera fase de análisis, la segunda fase de diseño, la tercera fase de desarrollo y por último, la fase de documentación.

- **Fase 1: Análisis del proyecto**

En esta primera fase, se realiza un análisis sobre la aplicación que se quiere desarrollar, dónde se investiga y decide las tecnologías que se van utilizar en el TFG. Además, se efectúa una búsqueda del software y las librerías que necesitaremos para la posterior implementación.

Es en esta fase se decide programar en lenguaje Java, dado que las librerías externas que encontramos, y que se han usado en el TFG, se pueden importar de manera sencilla en Java mediante el uso de dependencias Maven.

Cómo se va desarrollar en Java, se elige el IDE que se va a usar para desarrollar. Al principio, se usa Eclipse, pero más tarde, se pasa a usar IntelliJ porque es un IDE muy intuitivo y nos permite integración con Git desde el propio programa.

- **Fase 2: Diseño**

La segunda fase del proyecto, denominada fase de diseño, se emplea en realizar una serie bocetos de cada parte de la aplicación, diagramas de flujo y decidir la arquitectura que va a tener la herramienta.

En ella, se toma la decisión de utilizar el patrón MVC, explicado en el apartado 4.1. de este documento

También, en esta fase, se define la nueva funcionalidad “Recomendación orientada a empresa” y cómo será su implementación.

- **Fase 3: Desarrollo**

Esta fase es la más extensa en tiempo y la más compleja, ya que es en la que se implementan todas las funcionalidades de la aplicación. Internamente, esta fase, se divide en cinco “sub-fases”:

- Flujo de navegación

Se desarrolla el flujo que sigue el usuario a través de los diferentes menús de la herramienta.

- Extracción de tweets

Es probablemente la funcionalidad más compleja, en cuanto a desarrollo, y la que más tiempo ha llevado. Como su nombre indica, se encarga de extraer datos de twitter (tweets), para que los almacenemos en nuestra herramienta y se puedan analizar posteriormente.

Fue necesario tomar algunas decisiones como definir un objeto intermedio para poder convertir los *Status*, que devuelve Twitter, en los objetos *Tweet*, definidos en el proyecto. También qué información guardábamos de los tweets o incluso el idioma de estos.

- **Análisis de sentimiento**

Esta sub-fase contempla la implementación de la funcionalidad que permite analizar los tweets almacenados y generar como resultado un sentimiento sobre cada tweet.

Además de guardar en base de datos los resultados, se decide generar un fichero CSV, por si se quiere comprobar en algún otro momento.

- **Recomendación de videojuegos**

Se integra la librería Mahout, que contiene la implementación de un sistema de recomendación basado en usuarios. Este sistema se nutre de un fichero CSV, por lo que la decisión tomada en la sub-fase anterior, nos ayuda con ello.

Del mismo modo que en el análisis de sentimiento, aparte de mostrar al usuario las recomendaciones de videojuegos, se decide persistir también los resultados en un fichero CSV.

- **Recomendación orientada a empresa**

Se implementa la idea definida en la fase de diseño.

- **Fase 4: Documentación**

Es la última fase del proyecto. En ella se recoge toda la información sobre el TFG y se deja reflejada en el presente documento.

En la Figura 6.1. se observa la duración de cada una de ellas, mediante un diagrama de Gantt.

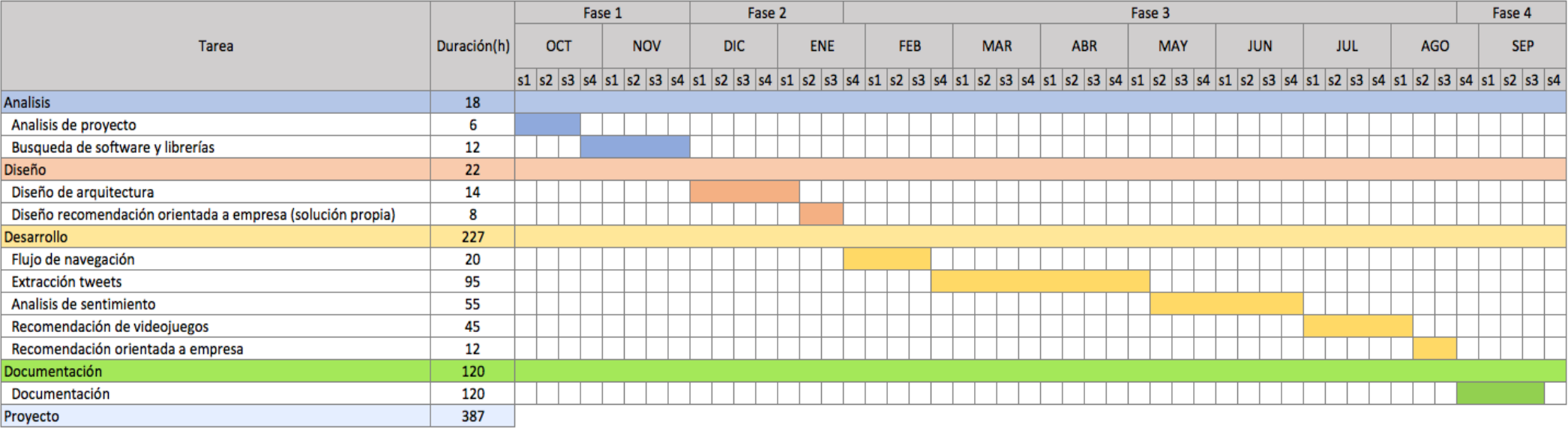


Figura 6.1 Diagrama de Gantt del proyecto

6.2 Presupuesto

En esta sección se muestra el presupuesto para llevar a cabo el TFG. Se desglosa en tres apartados principalmente: coste de material, coste de personal y el coste total del proyecto.

- **Costes de materiales**

Empezamos por los costes de los materiales, donde registramos los costes del hardware y el software que se ha utilizado en el proyecto.

- Apple Macbook Pro 13', 1200€
- API Aylien, 0€ (Licencia capa gratuita)
- Twitter4J, 0€ (Licencia gratuita)
- Apache Mahout, 0€ (Licencia gratuita)
- IntelliJ IDE, 0€ (Licencia estudiante)
- Microsoft Office 365, 0€ (Licencia uc3m)

Realizamos los cálculos de amortización utilizando la siguiente fórmula:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

Como resultado obtenemos la siguiente tabla:

Descripción	Coste(€)	%Uso dedicado al proyecto	Dedicación(meses)	Depreciación	Coste imputable(€)
MacBook Pro 13'	1200	100	12	60	240
API Aylien	0	100	4	12	0
Twitter4J	0	100	6	12	0
Apache Mahout	0	100	2	12	0
IntelliJ IDE	0	100	9	12	0
MS Office 365	0	100	1	12	0

Tabla 6-1 Cálculos de amortización

Por lo que el total en costes de materiales asciende a **240€**.

- **Costes de personal**

En esta sección calcularemos los costes del personal que ha intervenido en el TFG. Por un lado, tenemos a un graduado en ingeniería, un servidor, que ha sido el desarrollador del proyecto. Por otro lado, tenemos una ingeniera superior, Florina Almenares, mi tutora, que ha supervisado el TFG.

Se estima que el sueldo medio de un graduado de menos de 2 años de experiencia es de 24.000€ y el de un ingeniero senior en 45.000€. Por tanto, para un contrato de 40h/semana, obtendríamos 12,50€/h (graduado) y 23,43€/h (senior).

Tal y como se refleja en la planificación, las horas invertidas por el graduado son 387. En cuanto a la ingeniera senior, se estiman 50 horas entre reuniones presenciales, reuniones vía conferencia y tiempo de revisión del proyecto.

Nombre y apellidos	Categoría	Dedicación(h)	Sueldo (€/h)	Coste(€)
Alejandro Bermejo Vialás	Graduado en Ingeniería	387	12,50	4.837,50
Florina Almenares Mendoza	Ingeniera Superior	50	23,43	1.171,50

Tabla 6-2 Costes de personal

En total, los **costes de personal**, suponen **6.009€**.

- **Presupuesto total**

Ahora que se han realizado los cálculos para los costes de personal y costes de material, procedemos a obtener el presupuesto total del proyecto.

Descripción de coste	Coste(€)
Materiales	240
Personal	6.009
Indirectos (20%)	1.249,80
Total (IVA incl.)	7.498,80

Tabla 6-3 Presupuesto total

7 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

En este apartado se exponen tanto las conclusiones a las que se ha llegado como la posible trayectoria que puede tomar el proyecto, para llevar a cabo mejoras o modificaciones futuras en él.

7.1 Conclusiones

A lo largo de este documento, ha quedado reflejado cómo se lleva a cabo el desarrollo de un recomendador de videojuegos. Desde la toma de decisiones para la elección de las tecnologías usadas en el proyecto, hasta la forma en que se distribuyen las diferentes partes de código del mismo.

Una vez concluido todo el desarrollo y haber llevado a cabo las pruebas del recomendador, hemos podido comprobar que se cumplen los objetivos marcados al inicio del proyecto.

Para empezar, se ha desarrollado el recomendador de videojuegos de forma satisfactoria. La herramienta genera, como resultado del proceso, sugerencias acordes a las preferencias de los usuarios extraídos de Twitter. Aunque a lo largo de su desarrollo, han surgido dificultades a la hora de devolver dichas recomendaciones, ya que dependiendo de la forma en que se puede calcular la similitud entre usuarios, puede dar lugar a que no genere sugerencias. Para la distribución de los datos que se tiene en el TFG, los métodos que devuelven recomendaciones son Distancia euclídeana, Coseno y Tanimoto.

Por otro lado, la funcionalidad creada para recomendar usuarios en base a la categoría de los videojuegos, también ha sido implementada con éxito.

En cuanto a los objetivos y motivaciones personales, resaltar la satisfacción de haber creado una idea desde cero, sobre un tema que se desconocía, y haber sido capaz de desarrollarla por completo, de forma que se obtuvo como resultado un funcionamiento óptimo del recomendador.

7.2 Líneas de trabajo futuras

En esta sección, se proponen nuevas posibles vías de trabajo para mejorar algunas de las funcionalidades o por el mero hecho de añadir nueva funcionalidad a la herramienta.

- **Nueva Interfaz**

La aplicación desarrollada tiene una interfaz de usuario sencilla, cumple con su cometido, pero no es vistosa. Se podría implementar una interfaz web mediante JSP o tecnologías más actuales en el panorama frontend, como HTML5 y CSS3, que permiten crear una experiencia de usuario mejor que la que aporta la línea de comandos.

- **Plataforma web**

Esta segunda línea de trabajo, va de la mano con la descrita en el párrafo anterior. Podría llevarse a cabo el “plataformado” de la herramienta o la integración dentro de una plataforma web de Customer Experience, como nuevo producto. Esto le aportaría robustez a la aplicación y podría ser usada por diferentes clientes sin necesidad de instalaciones o versiones *on premise*. Por otro lado, como desarrollador, solo habría que mantener una plataforma, que sería a la que se irían aplicando los cambios, de modo que las correcciones o actualizaciones se centran en un solo punto.

- **Mejora en la búsqueda de hashtags**

Ahora mismo, la herramienta se nutre de los *hashtags* que aparecen en los tweets, pero una persona debe validarlos manualmente y decidir si son un videojuego o no. Sería muy interesante ver si se puede mejorar esto, de tal manera que se identificasen entidades de manera automática, con el uso de otras herramientas de *text mining*.

- **Nuevas entradas de información**

En el desarrollo actual, la única entrada de datos es la extracción de tweets. Se podrían implementar otras fuentes de datos, por ejemplo:

- Subir ficheros CSV con los datos a analizar.
- Integración de procesos automáticos en los que la herramienta recoja
 - información de otro servidor de almacenamiento.
- Una API en la que se reciba un JSON con los datos.

- **CI/CD**

En caso de desarrollar la plataforma web, sería interesante implementar un flujo de despliegues continuos. Por ejemplo, tener una instancia Jenkins que ejecute test del código, compile y lo suba a un *bucket* de AWS S3 y desde ahí que se ejecute el despliegue en las instancias.

- **Servicio REST de recomendación**

Desarrollar un servicio REST, que si se le manda un JSON con los datos de entrada (usuario, ítem, valoración), devuelva las recomendaciones sobre esos datos.

Bibliografía

- [1] Asociación Española de Videojuegos. (2018). *El videojuego en España*. [En línea] Disponible en: <http://www.aevi.org.es/la-industria-del-videojuego/en-espana/> [Último acceso 23 Sep. 2018].
- [2] Dle.rae.es. (2018). *Videojuego*. [En línea] Disponible en: <http://dle.rae.es/?id=bmnbNU7> [Último acceso 10 Sep. 2018].
- [3] Es.wikipedia.org. (2018). *Modelo–vista–controlador*. [En línea] Disponible en: <https://es.wikipedia.org/wiki/Modelo–vista–controlador> [Último acceso 15 Sep. 2018].
- [4] Es.wikipedia.org. (2018). *Videojuego*. [En línea] Disponible en: <https://es.wikipedia.org/wiki/Videojuego> [Último acceso 10 Sep. 2018].
- [5] Herranz, S. (2018). *La industria del videojuego en España bate récords de facturación*. [En línea] HobbyConsolas. Disponible en: <https://www.hobbyconsolas.com/reportajes/industria-videojuego-espana-bate-records-facturacion-270081> [Último acceso 23 Sep. 2018].
- [6] Es.wikipedia.org. (2018). *MySQL*. [En línea] Disponible en: <https://es.wikipedia.org/wiki/MySQL> [Último acceso 15 Sep. 2018].
- [7] Bannister, K. (2015). *Análisis de sentimiento: qué es y para qué se usa - Brandwatch*. [En línea] Brandwatch. Disponible en: <https://www.brandwatch.com/es/blog/analisis-de-sentimiento/> [Último acceso 18 Sep. 2018].
- [8] Es.wikipedia.org. (2018). *Análisis de sentimiento*. [En línea] Disponible en: https://es.wikipedia.org/wiki/Análisis_de_sentimiento [Último acceso 17 Sep. 2018].
- [9] Sites.google.com. (2018). 3. *Técnicas para Identificar Requisitos Funcionales y No Funcionales - Metodología Gestión de Requerimientos*. [En línea] Disponible en: <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales> [Último acceso 23 Sep. 2018].
- [10] Es.wikipedia.org. (2018). *Transferencia de Estado Representacional*. [En línea] Disponible en: https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional [Último acceso 16 Sep. 2018].
- [11] Codecademy. (2018). *What is REST? / Codecademy*. [En línea] Disponible en: <https://www.codecademy.com/articles/what-is-rest> [Último acceso 16 Sep. 2018].
- [12] Developer.twitter.com. (2018). *Standard search API*. [En línea] Disponible en: <https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html> [Último acceso 14 Sep. 2018].
- [13] Yamamoto, Y. (2018). *Twitter4J/Twitter4J*. [En línea] GitHub. Disponible en: <https://github.com/Twitter4J/Twitter4J> [Último acceso 20 Sep. 2018].
- [14] Yamamoto, Y. (2018). *Twitter4J - A Java library for the Twitter API*. [En línea] Twitter4j.org. Disponible en: <http://twitter4j.org/> [Último acceso 20 Sep. 2018].

- [15] Docs.aylien.com. (2018). *Text Analysis API Documentation*. [En línea] Disponible en: <https://docs.aylien.com/textapi/endpoints/#sentiment-analysis> [Último acceso 10 Sep. 2018].
- [16] Docs.aylien.com. (2018). *Text Analysis API Documentation*. [En línea] Disponible en: <https://docs.aylien.com/textapi/sdks#java-sdk> [Último acceso 10 Sep. 2018].
- [17] Es.wikipedia.org. (2018). *Java (lenguaje de programación)*. [En línea] Disponible en: [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci3n)) [Último acceso 23 Sep. 2018].
- [18] Java.com. (2018). *¿Qué es Java y para qué es necesario?*. [En línea] Disponible en: https://www.java.com/es/download/faq/whatis_java.xml [Último acceso 17 Sep. 2018].
- [19] En.wikipedia.org. (2018). *Write once, run anywhere*. [En línea] Disponible en: https://en.wikipedia.org/wiki/Write_once,_run_anywhere [Último acceso 17 Sep. 2018].
- [20] Es.wikipedia.org. (2018). *Maven*. [En línea] Disponible en: <https://es.wikipedia.org/wiki/Maven> [Último acceso 18 Sep. 2018].
- [21] Genbeta.com. (2011). *Introducción a Maven*. [En línea] Disponible en: <https://www.genbeta.com/desarrollo/introduccion-a-maven> [Último acceso 18 Sep. 2018].
- [22] Boe.es. (1999). BOE.es - *Documento BOE-A-1999-23750*. [En línea] Disponible en <https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750> [Último acceso 19 Sep. 2018].
- [23] Reglamento general de protección de datos de la UE (GDPR) - ESET. (2018). *Reglamento general de protección de datos de la UE (GDPR) - ESET*. [En línea] Disponible en: <https://gdpr.eset.es/#gdpr> [Último acceso 19 Sep. 2018].
- [24] Ibáñez, M.B. (2015). *Análisis de Requisitos: Diagramas de Caso de Uso*[Material de clase]. Software de Comunicaciones, Universidad Carlos III de Madrid, Leganés.
- [25] Boe.es. (2002). BOE.es - *Documento consolidado BOE-A-2002-13758*. [En línea] Disponible en: <https://www.boe.es/buscar/act.php?id=BOE-A-2002-13758> [Último acceso 19 Sep. 2018].
- [26] Lssi.gob.es. (n.d.). *Inicio - LSSI*. [En línea] Disponible en: <http://www.lssi.gob.es/paginas/Index.aspx> [Último acceso 19 Sep. 2018].
- [27] Boe.es. (1996). *BOE.es - Documento consolidado BOE-A-1996-8930*. [En línea] Disponible en: <https://www.boe.es/buscar/act.php?id=BOE-A-1996-8930> [Último acceso 19 Sep. 2018].
- [28] Owen, S., Anil, R., Dunning, T. and Friedman, E. (2011). *Mahout in action*. Shelter Island, NY: Manning Publications.
- [29] Sánchez, J., Laurentiu, A. and San Gabino, D. (2014). *easyMahout : entorno de ejecución de algoritmos inteligentes de Mahout para Hadoop y Big Data* - E-Prints

Complutense. [En línea] Eprints.ucm.es. Disponible en: <https://eprints.ucm.es/26534/> [Último acceso 4 Sep. 2018].

[30] Concepto.de. (2017). *¿Qué es Red Social? - Concepto, Definición y Características*. [En línea] Disponible en: <https://concepto.de/redes-sociales/> [Último acceso 9 Sep. 2018].

[31] Santiago, I. (2018). *Redes Sociales: Qué Son, Para Qué Sirven y Cómo Sacarlas Provecho*. [En línea] Ignacio Santiago. Disponible en: <https://ignaciosantiago.com/redes-sociales-que-son-para-que-sirven-como-usarlas/#que-son> [Último acceso 10 Sep. 2018].

[32] Histinf.blogs.upv.es. (2011). *Redes Sociales – Historia de la Informática*. [En línea] Disponible en: <http://histinf.blogs.upv.es/2011/12/20/redes-sociales/> [Último acceso 11 Sep. 2018].

[33] Es.wikipedia.org. (2018). *Twitter*. [En línea] Disponible en: <https://es.wikipedia.org/wiki/Twitter> [Último acceso 12 Sep. 2018].

[34] Mejía, J. (2018). *Estadísticas de redes sociales 2018: Usuarios de Facebook, Twitter, Instagram, YouTube, LinkedIn, Whatsapp y otros*. [En línea] Blog Marketing Digital, Social Media y Transformación Digital | Juan Carlos Mejía Llano. Disponible en: https://www.juancmejia.com/marketing-digital/estadisticas-de-redes-sociales-usuarios-de-facebook-instagram-linkedin-twitter-whatsapp-y-otros-infografia/#5_Usuarios_activos_de_Twitter [Último acceso 12 Sep. 2018].

[35] Es.wikipedia.org. (2018). *Sistema de recomendación*. [En línea] Disponible en: https://es.wikipedia.org/wiki/Sistema_de_recomendación [Último acceso 23 Sep. 2018].

[36] Villalonga, R. (2017). *Sistemas recomendadores basados en contenido – Rodrigo Villalonga – Medium*. [En línea] Medium. Disponible en: <https://medium.com/@rvillalongar/sistemas-recomendadores-basados-en-contenido-ece0227e7005> [Último acceso 20 Sep. 2018].

[37] Villalonga, R. (2017). *Filtrado Colaborativo (FC) basado en el usuario – Planeta Chatbot : todo sobre los Chatbots y la Inteligencia Artificial*. [En línea] Planeta Chatbot : todo sobre los Chatbots y la Inteligencia Artificial. Disponible en: <https://planetachatbot.com/filtrado-colaborativo-fc-basado-en-el-usuario-f13926ffdfac> [Último acceso 21 Sep. 2018].

[38] González, A. (2015). *Sistemas de recomendación de contenido con Machine Learning -*. [En línea] Cleverdata.io. Disponible en: <https://cleverdata.io/sistemas-recomendacion-machine-learning/> [Último acceso 23 Sep. 2018].

[39] Uv.es. (n.d.). *3.1 Coeficiente de Pearson*. [En línea] Disponible en: https://www.uv.es/webgid/Descriptiva/31_coeficiente_de_pearson.html [Último acceso 23 Sep. 2018].

[40] Es.wikipedia.org. (2018). *Coeficiente de correlación de Pearson*. [En línea] Disponible en: https://es.wikipedia.org/wiki/Coeficiente_de_correlación_de_Pearson [Último acceso 22 Sep. 2018].

[41]Json.org. (n.d.). *JSON*. [En línea] Disponible en: <http://www.json.org> [Último acceso 23 Sep. 2018].

ANEXO A. GLOSARIO

API	Application Performance Interface
AWS	Amazon Web Services
CSV	Comma Separated Values
DDBB	(data base) Base de Datos
ER	Entidad-Relación
GDPR	Reglamento General de Protección de Datos
LOPD	Ley Orgánica de Protección de Datos
LSSI	Ley de Servicios <i>¿¿¿</i>
RAE	Real Academia Española
S3	Simple Storage Service
TFG	Trabajo Fin de Grado
UC3M	Universidad Carlos III de Madrid